WHERE, WHAT, AND WHEN: STRUCTURED INTRA-RESPONSE ADVANTAGE REDISTRIBUTION FOR LLM REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards has substantially improved large-language-model reasoning, yet Group Relative Policy Optimization (GRPO) typically applies the same response-level advantage to all tokens in a sampled response. We introduce Where–What–When (WWW), a structured intra-response advantage redistribution framework that retains GRPO's group-relative advantage estimator while differentiating credit across reasoning segments. Where supplies a structural position prior, What uses segment-level surprisal evidence to select candidate segments, and When uses their disagreement to adapt the intervention strength and selectively invoke Beam–Monte Carlo Verification (BMCV) for outcome-sensitive calibration. Response-wise mean normalization controls the average weight scale before the redistributed advantage is used in the clipped policy objective. On Llama-3.2-3B-Instruct, WWW improves the best observed accuracy over our matched GRPO baseline by 2.20–3.34 percentage points across GSM8K, MATH500, AIME24, and AIME25. On Qwen2.5-7B-Instruct, within the first 500 training steps, WWW improves the best observed accuracy over GRPO by 2.84–6.70 points across the same benchmarks. The lightweight WWW variant without BMCV adds only 1.99% end-to-end step-time overhead, while Full WWW adds 12.69%. We will release our code and model weights.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has become an effective paradigm for improving the reasoning capabilities of large language models. For tasks such as mathematical reasoning, the final answer provides a natural outcome-level supervision signal, enabling policy optimization without dense human annotation. Group Relative Policy Optimization (GRPO)(Shao et al., 2024) further simplifies this process by estimating a group-relative scalar advantage from multiple responses sampled for the same prompt, avoiding the need to train a separate critic or value model.

Despite this simplicity, the supervision provided by GRPO remains coarse within each generated response. For a sampled response y_i, the same scalar advantage $\hat{A}_i$ is typically applied to all valid response tokens during the policy update. Consequently, GRPO effectively distinguishes which sampled responses should be reinforced or suppressed, but does not explicitly distinguish how strongly different reasoning segments within the same response should contribute to the update. This motivates the problem we study in this work: *given an outcome-derived response advantage, how should its credit be redistributed within the response without introducing a separately trained critic or process reward model?*

Autoregressive generation suggests a possible relationship between reasoning position and uncertainty. Early reasoning decisions are made before the subsequent trajectory has been established, whereas later generation is conditioned on an increasingly developed reasoning context. This motivates using segment position as a low-cost prior for allocating credit within a response. However, uncertainty need not decrease monotonically with generation position, and later computations or corrections may also be consequential. We therefore use position as a structural prior rather than treating it as a direct measurement of token uncertainty or outcome importance.

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.

Position alone cannot distinguish reasoning segments with different content at similar structural positions. To complement this coarse prior, token surprisal provides a content-dependent signal derived from the likelihood of the actual generated tokens under the rollout policy. Where and What thus offer two different forms of evidence for differentiating reasoning regions: Where expresses a position-based preference, whereas What identifies segments containing locally unexpected generation choices. Neither signal directly establishes how changing a local reasoning decision would affect the final outcome. Evaluating alternative local continuations can provide more direct outcome-dependent evidence, but applying such verification to every reasoning segment would incur substantial additional generation cost.

We propose Where–What–When (WWW), a framework for structured intra-response advantage redistribution. The Where component assigns a positional prior to reasoning segments, while the What component uses segment-level surprisal evidence to identify candidates for content-dependent weighting. The When component compares the relative preferences induced by the Where and What components within each response. Their disagreement modulates the strength of content-based intervention and determines whether additional verification is considered. For candidates satisfying the disagreement gate and computational constraints, we selectively invoke Beam-Monte Carlo Verification (BMCV), which evaluates local alternative branches and uses variation in their downstream returns to calibrate segment-level weighting. Disagreement thus guides when additional computation is allocated to verification, rather than serving as direct evidence of a segment's causal importance.

The resulting signals are combined into token-dependent weights that redistribute the original group-relative scalar advantage within each response. We retain the GRPO reward, group-relative advantage estimator, policy ratio, clipping mechanism, and KL regularization, while replacing the scalar advantage used in the token-level policy objective with its reweighted counterpart. Response-wise mean normalization keeps the average token weight approximately one, reducing the confounding effect of globally rescaling the advantage. The framework therefore follows a compute-allocation principle: lightweight structural and content-dependent signals are used broadly, whereas additional outcome-sensitive verification is performed selectively.

Our contributions are summarized as follows:

- We formulate fine-grained reasoning credit assignment in GRPO as **structured intra-response advantage redistribution**, allowing different reasoning regions to contribute differently to the same outcome-derived response credit without training an additional critic.
- We propose the **Where–What–When framework**, which combines a structural position prior, surprisal-based content evidence, and disagreement-guided intervention into a unified mechanism rather than treating them as independent token-weighting heuristics.
- We introduce **selective outcome-sensitive verification**, where Where–What disagreement determines when additional Beam–Monte Carlo computation is allocated, and branch-level outcome variation is used to calibrate rather than blindly amplify candidate segment weights.
- On Llama-3.2-3B-Instruct, WWW improves the best observed accuracy over our matched GRPO baseline by 2.20–3.34 percentage points across GSM8K, MATH500, AIME24, and AIME25. Within the first 500 training steps on Qwen2.5-7B-Instruct, the best observed gains range from 2.84 to 6.70 points. The lightweight variant without BMCV adds only 1.99% end-to-end step-time overhead.

2 Related Work

2.1 Reinforcement Learning with Verifiable Rewards

Reinforcement learning with verifiable rewards (RLVR) has become a widely used paradigm for improving mathematical and logical reasoning in large language models. GRPO was introduced in DeepSeekMath (Shao et al., 2024) as a critic-free variant of PPO that estimates a group-relative scalar advantage from multiple responses sampled for the same prompt. Its simplicity and reduced memory requirements make GRPO particularly attractive for reasoning post-training. However, the

resulting scalar advantage is typically shared across all valid tokens of a response, leaving fine-grained intra-response credit assignment unresolved.

Our work retains the standard GRPO reward and group-relative advantage estimator and instead studies how this outcome-derived scalar credit should be redistributed within a response.

2.2 Fine-Grained Credit Assignment for LLM Reasoning

Several recent methods seek to improve the granularity of credit assignment without relying on a conventional learned critic. GRPO-λ(Parthasarathi et al., 2025) introduces a critic-free approximation to λ-return credit assignment using token-level log-probabilities. TEMPO(Tran et al., 2025) constructs prefix trees from sampled responses and derives non-parametric prefix values to provide branch-aware corrections at decision points. Counterfactual credit assignment estimates span importance by masking reasoning spans and measuring changes in answer probability (Khandoga et al., 2026). More recently, OPPO(Li et al., 2026) accumulates oracle-conditioned likelihood evidence along the trajectory to obtain token-level estimates of eventual success without training a separate value network.

R^3L (Shi et al., 2026) localizes optimization to pivotal divergent suffixes after reflection-guided retry, while SORT (Le et al., 2026) uses plan-conditioned token-probability changes to recover selective learning signals from all-wrong rollout groups. Both methods introduce additional mechanisms for selective credit assignment through reflection-guided retry or reference-guided auxiliary updates, whereas WWW redistributes an existing GRPO response advantage using structural, surprisal-based, and disagreement signals.

Our approach differs in that it does not attempt to directly estimate a token-level value or causal contribution everywhere. Instead, it first constructs lightweight structural and surprisal-based signals over reasoning segments, and uses their disagreement to decide where stronger outcome-sensitive verification is warranted.

2.3 Process Supervision and Advantage Shaping

Process supervision provides another route toward finer-grained learning signals. CAPO(Xie et al., 2025) uses a general-purpose LLM as a generative process reward model to produce step-wise critiques and token-level rewards. PROF(Ye et al., 2025) combines outcome and process information through consistency-based sample filtering rather than directly blending their reward signals.

Other work modifies the optimization signal itself. VERL(Huang et al., 2026) studies exploration and exploitation in semantic hidden-state space and shapes the RL advantage using effective-rank-based signals and their temporal dynamics.

In contrast, our base intervention requires neither a separately trained process reward model nor dense auxiliary supervision. The What component reuses policy likelihoods already available from the rollout, while selective outcome-sensitive verification is reserved for a small subset of high-disagreement candidates.

2.4 Uncertainty Signals and Selective Verification

Model-internal confidence signals have been used to guide selective computation during language-model reasoning. SpecGuard (Purohit et al., 2026) combines log-probability-based confidence with an attention-based signal to guide step-level verification in speculative decoding.

Our setting differs from inference-time verification: we use token surprisal from the rollout policy to select candidate reasoning segments for advantage redistribution during RL training. Surprisal describes the likelihood of a generated token under the policy, whereas the effect of a local reasoning decision on the final reward requires outcome-dependent evidence. We therefore use surprisal for candidate selection and reserve additional verification for selected cases.

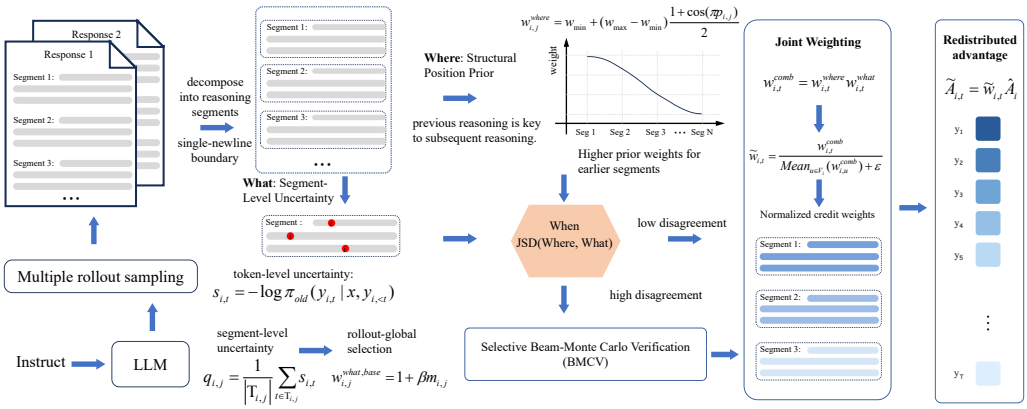

Figure 1: Overview of the proposed Where–What–When framework for structured intra-response advantage redistribution.

3 Preliminaries

Given a prompt x, GRPO samples a group of G responses $\{y_i\}_{i=1}^{G}$ from the rollout policy. Let r_i denote the outcome reward of response y_i. A group-relative scalar advantage can be computed by normalizing the rewards within the sampled group:

$$\hat{A}_i = \frac{r_i - \mathrm{Mean}_{j=1}^{G}(r_j)}{\mathrm{Std}_{j=1}^{G}(r_j) + \epsilon}, \tag{1}$$

where ϵ is a small constant for numerical stability. The resulting group-relative advantage enters the clipped GRPO policy objective. Ignoring implementation-specific averaging conventions, the policy optimization term can be written as

$$\mathcal{J}_{\mathrm{GRPO}}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\mathcal{V}_i|}\sum_{t\in\mathcal{V}_i}\min\Big(\rho_{i,t}(\theta)\hat{A}_i,\ \mathrm{clip}\left(\rho_{i,t}(\theta), 1-\epsilon_c, 1+\epsilon_c\right)\hat{A}_i\Big), \tag{2}$$

where $\rho_{i,t}(\theta) = \pi_\theta(y_{i,t} \mid x, y_{i,<t}) / \pi_{\mathrm{old}}(y_{i,t} \mid x, y_{i,<t})$ is the policy probability ratio. Here, π_θ denotes the current policy being optimized, while π_{old} denotes the rollout policy used to generate the sampled responses. Each policy assigns a conditional probability to token $y_{i,t}$ given the prompt x and preceding response tokens $y_{i,<t}$. Furthermore, $\mathcal{V}_i$ denotes the set of valid response tokens, and ϵ_c is the clipping coefficient.

In the standard GRPO formulation, all valid tokens within a sampled response share the same group-relative scalar advantage. This uniform assignment does not explicitly distinguish the relative contributions of different reasoning segments to the policy update. We therefore investigate how the original scalar advantage can be redistributed within a response while retaining the group-relative advantage estimator and the clipped policy optimization structure.

4 Method

4.1 Overview

Figure 1 illustrates the overall WWW pipeline. Given the group-relative advantage $\hat{A}_i$ of response i, we assign token-dependent weights and define

$$\tilde{A}_{i,t} = \tilde{w}_{i,t}\hat{A}_i, \tag{3}$$

where $\tilde{w}_{i,t}$ controls the relative contribution of token t to the same outcome-derived response credit. WWW retains the group-relative advantage estimator and clipped policy-optimization structure of GRPO, while changing how the scalar response advantage is distributed across tokens.

We operate at the reasoning-segment level by partitioning each response as

$$\mathcal{S}_i = (S_{i,1}, S_{i,2}, \ldots, S_{i,K_i}), \tag{4}$$

where $S_{i,j}$ is the j-th reasoning segment and K_i is the number of segments in response i. In the primary experiments, segment boundaries are defined by the tokenizer's exact single-newline token. Segment-level weights are broadcast to their corresponding response tokens.

4.2 WHERE: STRUCTURAL POSITION PRIOR

Earlier reasoning segments condition subsequent generation, motivating position as a low-cost structural prior rather than a direct estimate of outcome importance. We assign each segment a cosine-decayed positional weight:

$$w_{i,j}^{\text{where}} = w_{\min} + (w_{\max} - w_{\min}) \frac{1 + \cos(\pi \xi_{i,j})}{2}, \tag{5}$$

where $\xi_{i,j} = (j-1)/\max(K_i - 1, 1) \in [0,1]$ is the normalized position of segment $S_{i,j}$. We use $w_{\max} = 1.0$ and $w_{\min} = 0.3$; the positive lower bound retains nonzero weight for later segments. Segment-level weights are broadcast to their corresponding response tokens.

4.3 WHAT: SURPRISAL-BASED CONTENT EVIDENCE

Position alone cannot distinguish segments with different generated content. The What component therefore uses token surprisal as complementary policy-derived evidence. For generated token $y_{i,t}$,

$$s_{i,t} = -\log \pi_{\text{old}}(y_{i,t} \mid x, y_{i,<t}), \tag{6}$$

where π_{old} is the rollout policy.

For each eligible segment $S_{i,j}$, let $\mathcal{T}_{i,j}$ contain the top η_{tok} fraction of valid content tokens ranked by $s_{i,t}$. We define its surprisal score as

$$q_{i,j} = \frac{1}{|\mathcal{T}_{i,j}|} \sum_{t \in \mathcal{T}_{i,j}} s_{i,t}. \tag{7}$$

We use $\eta_{\text{tok}} = 0.2$ so that localized high-surprisal choices are not diluted by averaging over the entire segment; alternative ratios are evaluated in Appendix B.

Candidate selection is performed globally over the current rollout. Let $\mathcal{E}$ be the set of eligible segments and $M = |\mathcal{E}|$. We select the highest-scoring $\lceil \eta_{\text{seg}} M \rceil$ segments by $q_{i,j}$, with $\eta_{\text{seg}} = 0.2$. Accordingly, $m_{i,j} \in \{0,1\}$ indicates whether $S_{i,j}$ is selected; because selection is rollout-global, some responses may contain no selected segment.

The base What weight is

$$w_{i,j}^{\text{what,base}} = 1 + \beta m_{i,j}, \tag{8}$$

where $\beta = 0.3$. Thus, selected segments receive a base weight of 1.3, while unselected segments retain weight 1.

4.4 WHEN: DISAGREEMENT-GUIDED INTERVENTION

The Where and What components provide complementary segment-level signals. For each response, we normalize the Where weights $w_{i,j}^{\text{where}}$ and the continuous What scores $q_{i,j}$ over the same eligible segment support to obtain P_i^{where} and P_i^{what}, respectively. We measure their disagreement by

$$D_i = \text{JSD}_2\left(P_i^{\text{where}}, P_i^{\text{what}}\right), \qquad D_i \in [0,1], \tag{9}$$

where JSD_2 denotes the Jensen–Shannon divergence with base-2 logarithms. Responses with fewer than two eligible segments, or a degenerate zero-sum support, use $D_i = 0$.

The disagreement score modulates the additional What boost:

$$\begin{aligned} w_{i,j}^{\text{what,adaptive}} &= 1 + D_i \left(w_{i,j}^{\text{what,base}} - 1 \right) \\ &= 1 + \beta D_i m_{i,j}. \end{aligned} \tag{10}$$

Thus, unselected segments retain weight 1, while the boost of selected segments varies with D_i. When $D_i > \tau_D$ and the eligibility and budget constraints are satisfied, a selected segment may additionally enter BMCV verification.

4.5 Selective Beam-Monte Carlo Verification

Token surprisal identifies candidate reasoning segments, but does not directly measure whether changing a local reasoning decision would affect the final outcome. For selected candidates satisfying the disagreement gate and verification constraints, we use BMCV to evaluate the downstream consequences of local alternative branches.

For each verified candidate segment $S_{i,j}$, we establish a checkpoint before the candidate decision. We evaluate L branches, consisting of the original branch and the retained beam-search alternatives, and sample K_{MC} Monte Carlo continuations from each branch. Let $a \in \{1, \ldots, L\}$ index the evaluated branches and $k \in \{1, \ldots, K_{\text{MC}}\}$ index the continuations sampled from each branch. For candidate segment $S_{i,j}$, $R_{i,j}^{(a,k)}$ denotes the terminal reward of continuation k from branch a. The mean return of branch a is

$$Q_{i,j}^{(a)} = \frac{1}{K_{\text{MC}}} \sum_{k=1}^{K_{\text{MC}}} R_{i,j}^{(a,k)}. \tag{11}$$

We estimate the outcome sensitivity of the candidate decision using the variance of its branch mean returns:

$$C_{i,j} = \operatorname*{Var}_{a=1,\ldots,L} \left(Q_{i,j}^{(a)} \right), \tag{12}$$

where the variance is computed across the evaluated branches with population normalization.

We map this criticality estimate to a bounded calibration factor:

$$\phi_{i,j} = \text{clip} \left(\frac{C_{i,j}}{C_{\max}}, 0, 1 \right), \tag{13}$$

where $C_{\max} = \max\{(R_{\max} - R_{\min})^2/4,\ 10^{-8}\}$, and $R_{\min}$ and $R_{\max}$ are the configured reward bounds.

BMCV uses this factor to calibrate the disagreement-adaptive What weight of the verified candidate segment:

$$w_{i,j}^{\text{what,verified}} = 1 + \beta D_i \phi_{i,j}. \tag{14}$$

The factor $\phi_{i,j}$ scales only the additional boost βD_i. A factor of zero removes the boost, while a factor of one retains the pre-verification weight. Selected segments that do not undergo verification retain their disagreement-adaptive What weights.

4.6 Joint Advantage Redistribution

For the complete WWW framework, each selected segment initially receives the disagreement-adaptive What weight in Eq. 10. If a selected segment undergoes BMCV verification, this weight is replaced by the verified weight in Eq. 14. Unselected segments retain a What weight of 1. The resulting segment-level Where and What weights are broadcast to their corresponding response tokens. We first combine them multiplicatively to obtain the pre-normalization weight:

$$w_{i,t}^{\text{comb}} = w_{i,t}^{\text{where}} w_{i,t}^{\text{what}}. \tag{15}$$

To reduce changes in the overall scale of the advantage, we rescale the combined weights to have approximately unit mean across the valid tokens of each response:

$$\tilde{w}_{i,t} = \frac{w_{i,t}^{\text{comb}}}{\text{Mean}_{u \in \mathcal{V}_i} \left(w_{i,u}^{\text{comb}} \right) + \epsilon}, \tag{16}$$

Table 1: Mathematical reasoning results (%) on Llama-3.2-3B-Instruct. Our GRPO and WWW rows report the best observed accuracy for each benchmark under the same training/evaluation protocol. †: values reported by R^{3}L, whose math metric is trajectory-level average reward/solution accuracy. ‡: avg@16 values reported by SORT. External rows are shown for context and are not directly comparable across protocols; n/r denotes not reported. Bold highlights the better result within our matched GRPO–WWW pair.

Method	GSM8K	MATH500	AIME24	AIME25
GRPO†	66.40	39.80	n/r	n/r
R^{3}L†	68.80	40.80	n/r	n/r
GRPO‡	n/r	52.10	6.70	0.80
SORT‡	n/r	56.00	8.80	1.70
GRPO (ours)	74.60	47.60	13.33	3.33
WWW (ours)	**77.26**	**49.80**	**16.67**	**6.67**

where $\mathcal{V}_i$ denotes the valid response-token set.

Finally, the normalized token weight is applied to the original group-relative response advantage as in Eq. 3. The normalized weights have approximately unit mean, rather than unit sum, within each response. This reduces changes in the average advantage scale while allowing tokens to contribute differently to the policy update; it does not guarantee preservation of the gradient norm.

5 Experiments

5.1 Experimental Setup

Models and data. We use Llama-3.2-3B-Instruct for the primary experiments and Qwen2.5-7B-Instruct for the cross-model evaluation in Table 3. All methods are trained on MetaMath using the same preprocessing and reward construction within each matched comparison.

Training. All controlled variants use the same GRPO framework. Each rollout contains 512 prompts with $G = 5$ responses per prompt, yielding 2560 responses per step, and training runs for up to 1000 steps. Unless explicitly ablated, sampling, optimization, KL regularization, and response-length settings are shared across methods. Verified hyperparameters are provided in Appendix A.

Evaluation. We report answer accuracy on GSM8K, MATH500, AIME24, and AIME25. GSM8K is also used for validation during training. Table 1 reports the best observed accuracy for each benchmark over our GRPO and WWW training runs; fixed-step comparisons are reported separately in Table 2.

Compared variants. Starting from GRPO, we evaluate Where-only and What-only controls, their combination (Where + What), disagreement-guided weighting without BMCV (WWW w/o BMCV), and Full WWW.

5.2 Main Results

Table 1 compares our matched GRPO and WWW runs and provides R^{3}L (Shi et al., 2026) and SORT (Le et al., 2026) as reported external references. Because the external methods use different training and evaluation protocols, their values are not treated as strictly matched comparisons.

Under our matched protocol, WWW improves over GRPO by 2.66, 2.20, 3.34, and 3.34 percentage points on GSM8K, MATH500, AIME24, and AIME25, respectively. The gains are therefore observed across both routine and harder mathematical reasoning benchmarks, although the external rows in Table 1 should be interpreted only within their own published protocols.

Table 2: Component ablation on Llama-3.2-3B-Instruct. GSM8K validation accuracy (%) is reported at fixed training steps.

Method	Where	What	When	BMCV	500	1000
GRPO	–	–	–	–	61.64	34.04
Where	✓	–	–	–	74.45	74.00
What	–	✓	–	–	73.77	62.62
Where + What	✓	✓	–	–	73.84	**76.04**
WWW w/o BMCV	✓	✓	✓	–	74.78	75.51
Full WWW	✓	✓	✓	✓	**76.72**	74.60

Table 3: Best observed Qwen2.5-7B-Instruct accuracy (%) within the first 500 training steps. Each benchmark reports its best observed value within the common 500-step training budget.

Method	GSM8K	MATH500	AIME24	AIME25
GRPO	84.86	69.42	16.70	10.00
Full WWW	**88.10**	**72.26**	**20.00**	**16.70**

5.3 Component Ablations

Table 2 isolates the proposed components at matched checkpoints.

At step 500, all redistribution variants outperform GRPO and Full WWW achieves the highest accuracy, improving GRPO by 15.08 points. BMCV adds 1.94 points over WWW without BMCV at this checkpoint. The ordering changes at step 1000: Where + What reaches 76.04%, followed by WWW without BMCV at 75.51% and Full WWW at 74.60%, whereas GRPO falls to 34.04%. Thus, the effects of disagreement-guided intervention and BMCV are checkpoint-dependent rather than uniformly additive throughout training. Additional design controls are reported in Appendix B.

5.4 Training Dynamics

The fixed-checkpoint results in Table 2 show a clear difference in late-stage behavior. Vanilla GRPO degrades strongly by step 1000, while all redistribution variants retain substantially higher validation accuracy. We therefore report both fixed checkpoints and best-observed benchmark results instead of relying on a single selected training step.

5.5 Cross-Model Evaluation

To examine transfer beyond the primary 3B setting, we train GRPO and Full WWW on Qwen2.5-7B-Instruct under the same 500-step training budget and report the best observed accuracy for each benchmark within that horizon.

Within the common 500-step budget, the best observed WWW accuracy exceeds GRPO by 3.24, 2.84, 3.30, and 6.70 percentage points on GSM8K, MATH500, AIME24, and AIME25, respectively. The result provides evidence that the redistribution mechanism transfers to a larger model, while not requiring the best values to occur at the same checkpoint.

5.6 Computational Efficiency

We compare end-to-end step time under matched configurations.

The lightweight WWW components add only 1.99% end-to-end step-time overhead over GRPO. Selective BMCV raises the total overhead to 12.69%, showing that most additional cost comes from outcome-sensitive verification rather than the Where–What–When weighting itself.

Table 4: End-to-end training efficiency on Llama-3.2-3B-Instruct.

Method	Time/step (s)	Overhead
GRPO	654	0.00%
WWW w/o BMCV	667	+1.99%
Full WWW	737	+12.69%

6 Discussion

WWW is deliberately conservative in what it changes in GRPO. It retains the group-relative response advantage and modifies only how that credit is used within a response. The three sources of evidence operate at different costs and resolutions: position provides a deterministic structural prior, surprisal provides content-dependent evidence from the rollout policy, and BMCV is reserved for selected candidates that warrant an outcome-sensitive check. This coarse-to-fine design separates inexpensive screening from more costly verification while keeping the intervention interpretable at the reasoning-segment level.

The experimental protocol follows the same separation of roles. Matched GRPO controls are used for the main comparison, fixed checkpoints are reported for component analysis, and best-observed scores are used only where the corresponding table explicitly states that protocol. Results taken from prior work are reported as external references rather than treated as matched comparisons. The present study focuses on mathematical reasoning and newline-based segmentation; additional segmentation controls and design choices are reported in Appendix B.

7 Conclusion

We study fine-grained credit assignment in GRPO through structured intra-response advantage redistribution. WWW combines a structural position prior, surprisal-based content evidence, disagreement-guided intervention, and selective BMCV while retaining the original response-level group-relative advantage estimator. On Llama-3.2-3B-Instruct, WWW improves the best observed accuracy over our matched GRPO baseline from 74.60 to 77.26 on GSM8K, 47.60 to 49.80 on MATH500, 13.33 to 16.67 on AIME24, and 3.33 to 6.67 on AIME25. At step 500, Full WWW reaches 76.72% GSM8K validation accuracy versus 61.64% for GRPO, while the component ordering changes later in training, indicating that disagreement-guided intervention and BMCV are checkpoint-dependent rather than uniformly additive. On Qwen2.5-7B-Instruct, within the first 500 training steps, Full WWW also improves the best observed accuracy over GRPO by 2.84–6.70 percentage points across the four benchmarks. The lightweight WWW variant without BMCV incurs only 1.99% step-time overhead, and Full WWW incurs 12.69%.

Future work will study adaptive position and disagreement schedules, budget-aware BMCV, and learned or data-dependent reasoning segmentation. Extending the evaluation to additional model families, random seeds, and non-mathematical reasoning tasks will further clarify the conditions under which intra-response redistribution is most effective.

AI use statement

Generative AI tools were used to assist with literature discovery, research ideation, implementation and experiment discussions, figure design, and the drafting and polishing of portions of the manuscript. AI tools were also used to help inspect code, training logs, and experimental configurations. All methodological decisions, implementations, experimental results, analyses, citations, and claims were reviewed and verified by the authors. The authors take full responsibility for the content of this work.

Reproducibility statement

We provide the mathematical definition of each component of the proposed method in Section 4. Section 5.1 describes the main training and evaluation protocol, and Appendix A reports the verified

primary training and BMCV configuration. Additional design-choice results are reported in Appendix B. We intend to provide anonymized code and configuration files as supplementary material for review and will release the code and model weights publicly.

REFERENCES

Fanding Huang, Guanbo Huang, Xiao Fan, Yi He, Xiao Liang, Xiao Chen, Qinting Jiang, Faisal Nadeem Khan, Jingyan Jiang, and Zhi Wang. Semantic-space exploration and exploitation in RLVR for LLM reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 38402–38449, 2026. doi: 10.18653/v1/2026.findings-acl.1915.

Mykola Khandoga, Rui Yuan, and Vinay Kumar Sankarapu. Beyond uniform credit: Causal credit assignment for policy optimization. *arXiv preprint arXiv:2602.09331*, 2026.

Duc Anh Le, Tien-Phat Nguyen, Thien Huu Nguyen, Linh Ngo Van, and Trung Le. Selective off-policy reference tuning with plan guidance. *arXiv preprint arXiv:2605.11505*, 2026.

Yu Li, Rui Miao, Tian Lan, and Zhengling Qi. OPPO: Bayesian value recursion for token-level credit assignment in llm reasoning. *arXiv preprint arXiv:2605.21851*, 2026.

Prasanna Parthasarathi, Mathieu Reymond, Boxing Chen, Yufei Cui, and Sarath Chandar. GRPO-λ: Credit assignment improves llm reasoning. *arXiv preprint arXiv:2510.00194*, 2025.

Kiran Purohit, Ramasuri Narayanam, and Soumyabrata Pal. From tokens to steps: Verification-aware speculative decoding for efficient multi-step reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 17457–17471, 2026. doi: 10.18653/v1/2026.findings-acl.864. URL https://aclanthology.org/2026.findings-acl.864/.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Weijie Shi, Yanxi Chen, Zexi Li, Xuchen Pan, Yuchang Sun, Jiajie Xu, Xiaofang Zhou, and Yaliang Li. R^3L: Reflect-then-Retry Reinforcement Learning with Language-Guided Exploration, Pivotal Credit, and Positive Amplification. *arXiv preprint arXiv:2601.03715*, 2026.

Hieu Tran, Zonghai Yao, and Hong Yu. Exploiting tree structure for credit assignment in rl training of llms. *arXiv preprint arXiv:2509.18314*, 2025.

Guofu Xie, Yunsheng Shi, Hongtao Tian, Ting Yao, and Xiao Zhang. CAPO: Towards enhancing llm reasoning through verifiable generative credit assignment. *arXiv preprint arXiv:2508.02298*, 2025.

Chenlu Ye, Zhou Yu, Ziji Zhang, Hao Chen, Narayanan Sadagopan, Jing Huang, Tong Zhang, and Anurag Beniwal. Beyond correctness: Harmonizing process and outcome rewards through rl training. *arXiv preprint arXiv:2509.03403*, 2025.

A ADDITIONAL IMPLEMENTATION DETAILS

A.1 TRAINING CONFIGURATION

A.2 WHERE AND WHAT CONFIGURATION

The default Where component uses the tokenizer's exact single-newline token (token id 198) to define reasoning-segment boundaries and a cosine position prior with $w_{\max} = 1.0$ and $w_{\min} = 0.3$. For What, the segment score aggregates the top $\eta_{\text{tok}} = 0.2$ fraction of token surprisal values, and candidate selection uses the rollout-global top $\eta_{\text{seg}} = 0.2$ fraction of eligible segments. We compared $\eta_{\text{tok}} \in \{0.1, 0.2, 0.3\}$ and use 0.2 because it achieved the best validation performance among the tested ratios.

Table 5: Key training configuration for the primary 3B experiments.

Configuration	Value
Base model	Llama-3.2-3B-Instruct
Training data	MetaMath
Rollout batch size	512 prompts
Responses per prompt	5
Actor global batch size	128 prompts
Validation batch size	1024
Maximum training steps	1000
Learning rate	1×10^{-6}
Optimizer	AdamW
Weight decay	0.01
LR scheduler	constant
Advantage estimator	GRPO
KL coefficient	0.01
PPO epochs	1
Max. prompt / response length	1024 / 1024
Rollout temperature / Top-p	1.0 / 1.0

Table 6: Default disagreement and BMCV configuration used in the primary run.

Configuration	Value
Disagreement metric	base-2 Jensen–Shannon divergence
Disagreement threshold τ_D	0.30
What boost coefficient β	0.30
BMCV start step	200
Candidate prompt ratio	0.05
Maximum candidate prompts / step	16
Beam width	3
Retained alternatives	2
Beam max. tokens	64
MC continuations / branch	2
MC max. tokens	384
Minimum candidate-segment tokens	4
Minimum remaining tokens	24
Reward spread required	yes
Reward range	$[0, 1]$

A.3 When and BMCV Configuration

B Additional Ablations and Design Choices

B.1 Segmentation Boundary

The primary Where configuration uses the tokenizer's exact single-newline token to define reasoning segments. We additionally tested a double-newline boundary under the same positional intervention. The best validation accuracy decreases from 76.04% with single-newline segmentation to 75.21% with double-newline segmentation. We therefore use the finer single-newline boundary in the main experiments. This comparison also indicates that segmentation granularity is a nontrivial design choice rather than a purely syntactic preprocessing decision.

B.2 Reverse-Position Control

To test whether the gain from Where can be explained by arbitrary position-dependent reweighting, we reverse the positional preference so that later segments receive larger prior weights. Across the first five evaluation checkpoints within the first 500 optimization steps, the reverse-position control obtains 72.74, 72.48, 63.46, 49.58, and 28.05% GSM8K validation accuracy, respectively. The

pronounced late decline is consistent with the directional choice used by the default Where prior rather than position weighting alone.

B.3 Token-Selection Ratio

For the segment-level surprisal aggregation in What, we evaluated $\eta_{\text{tok}} \in \{0.1, 0.2, 0.3\}$. The 0.2 setting yielded the best validation performance and is used in all main experiments. The reported main method additionally uses rollout-global segment selection with $\eta_{\text{seg}} = 0.2$.

C Additional Evaluation Results

C.1 Cross-Model Evaluation

Within the first 500 training steps on Qwen2.5-7B-Instruct, the best observed GRPO accuracies are 84.86, 69.42, 16.70, and 10.00% on GSM8K, MATH500, AIME24, and AIME25, respectively. Full WWW reaches 88.10, 72.26, 20.00, and 16.70%, corresponding to improvements of 3.24, 2.84, 3.30, and 6.70 percentage points. These values use the same 500-step training budget but report the best observed accuracy for each benchmark within that horizon rather than a single matched checkpoint.