Not All Tokens Earn the Reward: Contrastive Credit Assignment for Production Detection

Tejas Alisyam **Prateek Sircar** **Deepak Gupta**
Amazon
{tejasyam, sircarp, dgupt}@amazon.com

Abstract

Seller-authored product titles often contradict their own bullet points (a title reading "128GB" against a bullet claiming "512GB"), which misleads shoppers and drives returns. At catalog scale, detection is precision-binding: even a small false-positive rate flags a huge absolute volume of good listings. We deploy a two-stage cascade on one national storefront of a large e-commerce marketplace: a compact 4B detector flags candidates and a strong reasoning model (DeepSeek R1) verifies them, reaching >99% precision while scoring all 36M listings. Verification is the expensive stage, so our goal is to fold it into the cheap model and run on the 4B detector alone; until then, every gain in the detector's recall catches more defects per verifier call and lifts coverage. We improve that first stage with Contrastive Token Weighting (CTW), a per-token credit-assignment method for GRPO. GRPO spreads one sequence-level advantage uniformly across a completion, even though only a few *pivotal* tokens decide correctness; CTW instead reads credit from a signal the reward already computes, the split of each sampled group into correct and incorrect completions, and upweights positions where a frozen scorer tells the two apart. The scorer can be a separate stronger teacher (CTW-T) or, when none is available, a frozen copy of the student itself (CTW-S, teacher-free); the teacher-free variant is the one we deploy, and it lifts the detector to 85.9 F1 (+2.2 over SFT, with the gain concentrated in recall, +3.4). A controlled math study shows the reward-induced partition, not the reweighting itself, is what drives the gain.

1 Introduction

Online marketplaces ask sellers to describe a product twice: a short title and a longer set of feature bullets. The two should agree, but at catalog scale they often do not: a title promising "128GB Storage" against a bullet advertising "512GB Internal Memory", or a polo sold as "Cotton" that a bullet lists as a cotton–polyester blend. Scanning a national storefront of a large e-commerce marketplace, we find that 2.74M of its 36M active listings (7.6%) carry a title–bullet contradiction of this kind. Each one is a small breach of trust at the moment a customer decides to buy, and the cost of a wrong call is lopsided: a missed contradiction reaches the customer, while a false flag only lands a good listing in a seller-facing review queue. At catalog scale even a small false-positive rate flags a huge absolute volume of good listings, so production precision must stay very high (we hold it above 99%) while recall still has to climb to catch real defects.

We meet both constraints today with a two-stage cascade: a compact 4B model flags candidate inconsistencies, and a strong reasoning model, DeepSeek R1 (DeepSeek-AI, 2025), verifies each flag. Verification is the expensive part: R1 is $\sim 16\times$ slower than the 4B detector on the same hardware and needs high-end datacenter GPUs, while the 4B model runs on commodity nodes. We would therefore like to collapse the cascade into a single 4B detector and retire R1; until then, every gain in the detector's recall lifts coverage while moving toward that endpoint. This paper is about improving that first stage with a better credit-assignment method for RL, and what we learned deploying it.

The verdict on each listing turns on a handful of *pivotal* tokens buried in otherwise interchangeable prose, so we train the detector with reinforcement learning on top of supervised fine-tuning (SFT) (Chu et al., 2025). The production verifier already supplies an exact binary reward, making Group Relative Policy Optimization (GRPO) (Shao et al., 2024) a natural fit. But GRPO applies one sequence-level advantage uniformly to every token, so of a $\sim$150-token output it credits the $\sim$10 decision tokens no more than the boilerplate. The obvious fix, concentrating learning where student

and teacher most disagree by KL divergence, gave us no lift, because KL tracks stylistic disagreement rather than reasoning (Section 2).

Our method, Contrastive Token Weighting (CTW), instead reads credit from a signal that GRPO uses only at sequence level: the split of each sampled group into correct and incorrect completions. A frozen scorer's per-token log-probabilities are compared across the two subsets, and positions where the log-probability gap is largest receive higher weight in the loss. The scorer can be a separate stronger teacher (CTW-T) or a frozen copy of the student itself (CTW-S, teacher-free); since our production reward model does not share the detector's tokenizer, we deploy CTW-S, and it trails CTW-T by little (Section 4).

This paper makes three contributions. **(i)** We deploy a CTW-S-trained 4B detector behind a reasoning-model verifier on a national storefront of a large e-commerce marketplace, reaching >99% precision while scoring all 36M listings and flagging 2.74M inconsistencies (Section 3). **(ii)** We introduce CTW, which derives per-token credit from the within-group correct/incorrect partition through a frozen scorer's log-probability contrast, motivated by a negative result (KL-based reweighting fails); its teacher-free variant CTW-S replaces the teacher with a frozen copy of the student, so no external model is needed at training time (Section 2). **(iii)** CTW-S lifts the deployed detector to 85.9 F1 (+2.2 over SFT, +3.4 recall); on MATH both CTW-T and CTW-S beat GRPO, DAPO, Dr. GRPO, and λ-GRPO (Yu et al., 2025; Liu et al., 2025c; Wang et al., 2025b), and a shuffled-partition control shows the reward-induced partition, not the reweighting, is the driver (Section 4).

2 Contrastive Token Weighting

Preliminaries (GRPO). Given a prompt x, GRPO (Shao et al., 2024) samples a group of G completions $\{y_1, \ldots, y_G\}$ from the policy π_θ, scores each with a reward r_i, and forms a group-relative advantage $A_i = (r_i - \text{mean}_j\, r_j)/(\text{std}_j\, r_j + \epsilon)$. The policy maximizes

$$\mathcal{L}_{\text{GRPO}} = -\frac{1}{G}\sum_{i=1}^{G} A_i \sum_{t=1}^{|y_i|} \log \pi_\theta(y_{i,t} \mid x, y_{i,<t}). \tag{1}$$

The advantage A_i is a scalar applied uniformly to all $|y_i|$ tokens, so it makes no *within-sequence* distinction, which is exactly the limitation CTW addresses.

A failed first attempt: student–teacher KL. The decision-critical tokens are a small minority, so a natural idea is to find them where the student and a stronger teacher disagree most: take the per-position KL divergence between their next-token distributions and use it as a multiplicative weight on the loss. This gave no statistically significant lift over vanilla GRPO. The high-KL positions turn out to be dominated by *stylistic* disagreement (phrasing, formatting, synonym choice) rather than the tokens that decide the answer, so KL measures a proxy that is not tied to the reward. This motivates CTW, which looks for a signal that is high exactly at the positions separating correct from incorrect *outcomes*.

Formulation. The verifiable reward splits the G completions of a group into a correct subset $\mathcal{C} = \{i : A_i > 0\}$ and an incorrect subset $\mathcal{I} = \{i : A_i \le 0\}$ (Figure 1 sketches the pipeline). CTW reads token credit straight from this split. A frozen scoring model π_T provides per-token log-probabilities

$$\ell_{i,t} = \log \pi_T(y_{i,t} \mid x, y_{i,<t}), \tag{2}$$

and we form a per-token contrastive signal that compares each completion against the opposing subset,

$$s_{i,t} = \begin{cases} \ell_{i,t} - \dfrac{1}{|\mathcal{I}|}\displaystyle\sum_{j\in\mathcal{I}} \ell_{j,t} & \text{if } i \in \mathcal{C}, \\ \dfrac{1}{|\mathcal{C}|}\displaystyle\sum_{j\in\mathcal{C}} \ell_{j,t} - \ell_{i,t} & \text{if } i \in \mathcal{I}. \end{cases} \tag{3}$$

For correct completions, $s_{i,t}$ is large when the scorer assigns higher log-probability to the produced token than to the matching tokens in incorrect completions; for incorrect completions, $s_{i,t}$ is large when the scorer is more confident about the correct completions' tokens at that position. We then normalize per sequence and turn the signal into multiplicative weights:

$$\begin{aligned} \bar{s}_{i,t} &= \frac{s_{i,t}}{\frac{1}{|y_i|}\sum_{t'} s_{i,t'} + \epsilon}, \\ w_{i,t} &= \text{clamp}\big(1 + \alpha(\bar{s}_{i,t} - 1),\, c_{\min}\big), \end{aligned} \tag{4}$$

with α=0.1 and $c_{\min}$=0.5 in all experiments. The clamp is one-sided: it floors weights at $c_{\min}$ so every token still contributes some gradient, but leaves

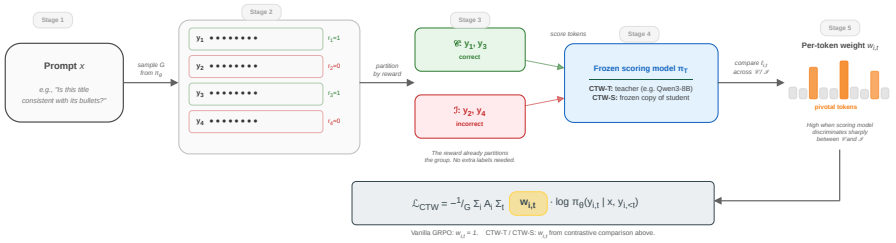


Figure 1: CTW pipeline. The reward partitions a GRPO group into correct ($\mathcal{C}$) and incorrect ($\mathcal{I}$); a frozen scorer (CTW-T: teacher, CTW-S: student copy) scores tokens across both subsets, and divergent positions receive higher $w_{i,t}$ in the policy loss.

them unbounded above so that strongly discriminating positions can be upweighted. When $\mathcal{C}$ or $\mathcal{I}$ is empty (all completions in a group share the same reward), the contrastive signal is undefined, and we set $w_{i,t} = 1$ for that group, recovering vanilla GRPO. This case follows directly from the advantage definition: with a binary reward, a unanimous group has $\mathrm{std}_j\, r_j = 0$, so every $A_i \to 0$ and, since $\mathcal{C} = \{i : A_i > 0\}$, $\mathcal{C}$ is empty; such groups therefore fall back to uniform weighting. In CTW-T the scorer π_T is a separate stronger frozen model; in the teacher-free variant CTW-S it is a frozen copy of the student itself, taken once from the SFT checkpoint and held fixed for the entire GRPO run (never refreshed), which keeps the contrastive reference stationary rather than chasing a moving policy and removes any external dependency.

Stability and cost. Two choices keep the signal stable: per-sequence mean-normalization (Eq. 4) removes completion-level scale bias, and averaging over the opposing subset (Eq. 3) cancels per-position vocabulary bias, leaving the tokens that genuinely separate correct from incorrect. The contrast is position-indexed, so it is most meaningful where prefixes align; normalization dampens misalignment but does not fully remove it, and in practice the weights concentrate on short, coherent regions (Appendix Figure 3). The method is also cheap: $w_{i,t}$ multiplies the per-token log-probability term in the GRPO objective (Eq. 1), training stays on-policy with no KL term, and the only added cost is one forward-only pass of the frozen scorer per completion (no backward pass or gradients; the weight computation is elementwise). Because each GRPO step is dominated by generating the G completions and the policy forward–backward, this scorer pass is a small fraction of step time: in our production setup it adds $\sim$12% wall-clock per training step. Algorithm 1 gives the full procedure.

Sparsity of the contrastive signal. The contrastive signal is defined only for groups that contain both a correct and an incorrect completion; in our production training run only $\sim$13% of groups meet this condition (the rest produce all-correct completions, since we train on SFT failure cases that the model is just learning to solve), and they fall back to uniform weights. Yet this 13% drives the entire improvement, because these are the prompts at the model's decision boundary, where it sometimes succeeds and sometimes fails and credit assignment matters most. CTW spends its gradient budget on the examples still being learned rather than on those already solved or hopeless.

3 Production System: Catalog Inconsistency Detection

We complement the controlled study in Section 4 with a full production deployment that differs in domain, task format, and model, sharing only GRPO's grouped-rollout structure and a verifiable binary reward, so it also tests whether the mechanism carries beyond reasoning benchmarks.

3.1 A precision-binding cascade

The cascade (Figure 2) reaches 99.1% precision on a manual audit of $\sim$2,000 flagged listings. Confirmed flags pass to a human-in-the-loop seller workflow; the model never auto-edits, so a residual error costs a seller-facing review, not a live catalog change.

3.2 Task and data

We frame detection as binary classification over a (title, bullet points) pair: the model writes a short reasoning trace explaining its comparison, then a tag (consistent / inconsistent). As the introduction notes, surface-form overlap is a poor proxy in both directions: telling a genuine conflict from an acceptable paraphrase or superset is what makes this

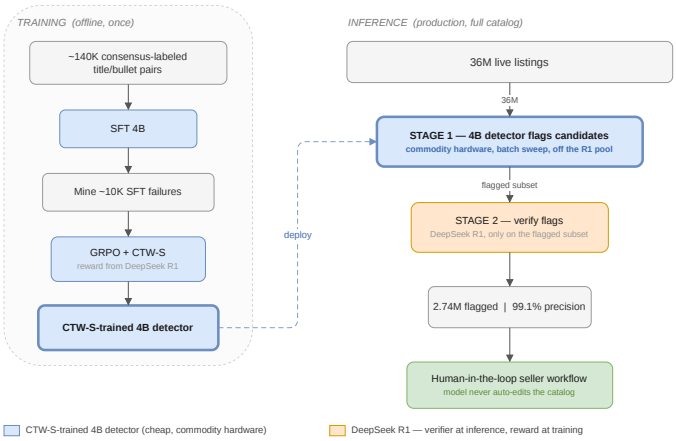


Figure 2: Production cascade. *Training (left):* consensus labels train a 4B SFT detector, refined by GRPO + CTW-S on its failures with DeepSeek R1 as reward. *Inference (right):* the deployed 4B detector (blue) flags all 36M listings; R1 (orange) verifies only the flagged subset, reaching 99.1% cascade precision before human review.

a credit-assignment problem rather than keyword matching.

Why not synthetic data. No labeled inconsistency dataset existed, so our first attempt bootstrapped one: take real (title, bullets) pairs and perturb a shared attribute to inject a contradiction ("Cotton"→"Bamboo Fiber", "Pack of 6"→"Pack of 3"). Three problems sank this. The pairs we assumed clean often already carried genuine inconsistencies, so the model's correct catches were scored as false positives; generation was expensive, several API calls per example; and single-slot perturbations cover only a sliver of the inconsistency space, teaching the model a narrow, overly strict notion of what counts rather than the full range. The lesson, which shaped everything downstream, is that the training labels had to come from real catalog items.

Labels via multi-model consensus. Without human annotators at the scale we needed, we label real listings by agreement among independent reasoning models. We curate a diverse gold seed with one strong model, then optimize detection prompts independently on two locally hosted reasoners that have systematically different failure modes (one over-flags synonyms, the other over-flags hierarchical relations). Requiring both to agree independently filters out each model's idiosyncratic false positives while keeping the contradictions both recognize, and a final verification prompt challenges each surviving flag; this multi-model agreement protocol is in the spirit of evidence-weighted multi-agent debate for more reliable LLM judgments (Verma et al., 2026). This yields ∼140K high-precision training labels (about 86% consistent / 14% inconsistent). We evaluate on a disjoint held-out set of 14K catalog-sampled pairs in a 90/10 split (90% consistent / 10% inconsistent, matching real-world prevalence). Labeled examples covering the conflict and paraphrase cases are in Appendix B (Table 4).

3.3 Training the first stage: SFT, then CTW on failures

We SFT Qwen3-4B-Instruct on the ∼140K consensus-labeled pairs, reaching 83.7 F1 on the held-out 14K set (Table 1). Companion work on this same detection task found that, for small detectors, knowledge distillation from a stronger model can outperform vanilla GRPO on the SFT model's residual failures (Alisyam et al., 2026); CTW instead narrows that gap by improving GRPO's credit assignment directly. We then add a GRPO stage (Chu et al., 2025) on the ∼10K cases SFT still gets wrong, calling R1 online to verify each rollout as the binary reward, so the detector improves by reinforcing its *own* correct rollouts rather than imitating R1's outputs. Since R1 does not share the detector's tokenizer, it cannot serve as the per-token scorer, so we apply the teacher-free CTW-S (Section 2). We report each method at its best held-out-F1 checkpoint (recall is measured at fixed precision ≈87, so it reflects more defects caught, not a precision–recall trade). Baselines are vanilla GRPO and λ-GRPO (Wang et al., 2025b) on the same checkpoint and failure subset; all RL methods share hyperparameters (Appendix A) and

differ only in loss weighting.

Table 1: First-stage detection on the held-out 14K set. RL rows are means over 3 seeds (std ≤ 0.24; significance in text); best per column among the deployable 4B models in bold.

Method	P	R	F1
Frontier commercial LLM (zero-shot)	44.8	94.0	60.7
SFT 4B (shared base)	86.4	81.2	83.7
+ Vanilla GRPO	86.9	82.8	84.8
+ λ-GRPO	87.0	82.9	84.9
+ **CTW-S (ours)**	**87.3**	**84.6**	**85.9**

3.4 First-stage results

Table 1 shows that both vanilla GRPO and λ-GRPO improve over SFT by roughly +1.1 F1 on these failure cases. CTW-S improves further to 85.9 F1 (+2.2 over SFT, +1.0 over λ-GRPO), with the gain concentrated in recall: +3.4 over SFT and +1.7 over λ-GRPO (84.6 vs. 82.9), so the detector catches more genuine inconsistencies at the same operating precision. Both the F1 and the recall gains over λ-GRPO are significant at $p < 0.001$ (Welch's t-test, $n = 3$ seeds); the precision difference is not (87.3 vs. 87.0, $p \approx 0.06$). A frontier model prompted zero-shot is no substitute for the trained first stage: a leading commercial reasoning model[1] reaches 94.0 recall but only 44.8 precision (60.7 F1), over-flagging so heavily that it would forward roughly half the catalog to the verifier, the opposite of what the cascade needs.

3.5 Deployment and impact

We deployed the cascade across the full positive-traffic catalog of one national storefront (36M active listings), flagging 2.74M (7.6%) as title–bullet inconsistencies. Serving all models with vLLM, the 4B detector sweeps the full catalog on commodity V100s; R1 verifies only the flagged subset, costing at least $\sim$450 node-hours (8$\times$H200) versus the $\sim$5,900 it would need as the sole detector (consistent with the $\sim$16$\times$ throughput gap).

Before the sale: conversion. Keeping only listings with enough traffic to estimate conversion (orders per thousand page views; $\sim$0.6M flagged, $\sim$7M consistent), the inconsistent-to-consistent ratio falls steadily with traffic on the primary storefront: near parity in the low-traffic deciles, dropping to 0.67 (33% lower) in the highest-traffic decile (Appendix Figure 4). A second national storefront repeats the pattern: 13% lower overall, falling to 0.76 (24% lower) in the top decile and 0.77 (23% lower) among listings with at least 10K page views.

After the sale: returns. The same partition predicts a second, independent outcome: an inconsistent page sets a wrong expectation, so the product is likelier to arrive "not as described" and be returned. Keeping only listings with enough orders to estimate a return rate, flagged listings are returned more: among listings with at least 10 orders the return rate is 5.46% for flagged versus 4.44% for consistent, a 23% relative increase that holds across order-volume floors from 10 to 100. Inconsistency thus costs the storefront twice, fewer sales before the purchase and more returns after it. Both signals are observational correlations, not causal fix effects (flagged listings may differ from consistent ones in other ways), so we read them as a lower bound on the problem's cost; a randomized fix-versus-holdback test is the planned next step. Flag rates vary by product line, highest for attribute-rich apparel (9.0%) and lowest for simply described consumables (3.6%; Appendix B).

4 Controlled Study: Math Reasoning

A single production run cannot isolate *why* CTW works, so we turn to public math-reasoning benchmarks for a controlled setting with published baselines and a shuffled-partition control. Two questions matter: whether the partition signal generalizes beyond the catalog task, and how much the teacher adds over the partition alone (the evidence for deploying the teacher-free variant).

4.1 Setup

Students are Qwen3-1.7B-Base and 4B-Base (full fine-tune); the frozen CTW-T teacher is Qwen3-8B-Base. We train on the Hard subset of the MATH training split (Hendrycks et al., 2021) and evaluate on GSM8K (Cobbe et al., 2021) and a held-out MATH test set (Hendrycks et al., 2021) (2-shot, Qwen-boxed). All methods share every hyperparameter (G=8, effective batch 32, lr 3×10^{-5}, 160 steps; CTW adds α=0.1, $c_{\min}$=0.5) and differ only in the loss weighting; we report the mean over 3 seeds at each method's best held-out-MATH checkpoint under the same selection rule. Baselines are vanilla GRPO (Shao et al., 2024), DAPO (Yu et al., 2025), Dr. GRPO (Liu et al., 2025c), λ-

[1] We report this baseline anonymously and omit the specific model name; it is a widely used frontier commercial LLM.

GRPO (Wang et al., 2025b), and the teacher-free CTW-S; Dr. GRPO is run at 1.7B and the remaining baselines at both scales (Appendix A).

4.2 Main results

Table 2: Accuracy (%) on math reasoning (Qwen3-8B-Base is the CTW-T teacher); mean over 3 seeds (std ≤ 0.42; significance in text), best per column per scale in bold.

Student	Method	GSM8K	MATH
1.7B	Zero-shot	68.99	32.40
	Vanilla GRPO	69.56	34.00
	DAPO	68.99	32.60
	Dr. GRPO	69.70	32.20
	λ-GRPO	69.86	33.10
	CTW-S (no teacher)	69.22	35.40
	CTW-T (8B teach.)	**70.28**	**35.90**
4B	Zero-shot	80.74	41.80
	Vanilla GRPO	81.62	42.68
	DAPO	80.89	42.20
	λ-GRPO	80.29	43.40
	CTW-S (no teacher)	80.95	43.50
	CTW-T (8B teach.)	**81.86**	**43.90**

We report relative improvement over matched baselines and focus on MATH, which has the most headroom. In Table 2 the aggregation-level baselines (GRPO, DAPO, Dr. GRPO, λ-GRPO) cluster together, while both CTW variants break out: at 1.7B, CTW-T vs. GRPO ($+1.90$) and CTW-S vs. GRPO ($+1.40$) are significant at $p < 0.001$ (Welch t-test, n=3), and the advantage holds at 4B (Figure 5 shows both separating from the cluster by step 60). Crucially, the teacher-free CTW-S trails CTW-T by only $+0.50$ ($p \approx 0.006$), suggesting that most of the gain comes from the partition itself rather than the teacher's capacity; Section 4.3 tests this directly.

4.3 The reward-induced partition is the driver

A natural alternative is that the gain comes simply from per-token heterogeneous weights breaking GRPO's uniform weighting, irrespective of how the two subsets are chosen. We test this by training CTW-T at 1.7B with the correct/incorrect partition randomly shuffled within each group, holding all else (teacher scoring, normalization, clamping, hyperparameters) identical. Shuffled CTW-T reaches 33.60% on MATH, within noise of vanilla GRPO (34.00%) and well below CTW-T's 35.90%; this 2.3-point gap is an order of magnitude larger than the per-method seed standard deviations ($\leq$0.42). Random partitioning recovers essentially none of the gain, so what matters is the reward-aligned partition, not heterogeneous weights per se. Together with CTW-S's near-parity to CTW-T, this is the evidence behind our goal of retiring the verifier: the partition carries the signal, so a frozen self-copy lets the first stage keep improving without a large model in the loop.

5 Related Work

Token-level credit in GRPO. Many methods reshape GRPO's uniform per-token signal: Dr. GRPO fixes a length-normalization bias (Liu et al., 2025c), GMPO uses a geometric-mean token average (Zhao et al., 2025), and λ-GRPO learns token weights (Wang et al., 2025b), while others source it from rollout or entropy statistics (Sun et al., 2025; Liu et al., 2025b; Lin et al., 2025; Tan et al., 2025), including high-entropy "forking" tokens (Wang et al., 2025a) and segment-level credit (Guo et al., 2025). All read credit from *policy-intrinsic* proxies (entropy, likelihood, rollout counts, learned parameters) and are evaluated only on math; CTW instead reads it off the reward-induced correct/incorrect partition, and is, to our knowledge, the first to derive token-level credit *during* RL training from that partition rather than from policy-intrinsic proxies, a separately trained process verifier, or inference-time pruning. Token-level preference methods need trained contrastive models and preference pairs rather than a frozen scorer and a binary reward: TIS-DPO contrasts a trained π^+/π^- pair (Liu et al., 2025a), with TDPO (Zeng et al., 2024), RTO (Zhong et al., 2025), and AlignDistil (Zhang et al., 2025) in the same family; process reward models add a separately trained step-level verifier (Lightman et al., 2023; Wang et al., 2023; Cui et al., 2025).

Self-distillation and production NLP. A concurrent line contrasts correct against incorrect rollouts within a group, mirroring our teacher-free CTW-S, via an oracle-conditioned likelihood ratio (Li et al., 2026b), a stop-gradient self-teacher (Li et al., 2026a), or a hidden-state Wasserstein distance (Chen et al., 2026); CTW's signal is instead a position-indexed log-probability *difference* from a single frozen scorer. On the application side, catalog-quality systems have used instruction-tuned evaluators (Joshi et al., 2025), data cleaners (Cheng et al., 2023), LLM ensembles (Fang et al., 2024), generative multimodal attribute extraction from product pages (Khandelwal

et al., 2023), and mixture-of-expert architectures for multi-source multimodal question answering (Verma et al., 2025), and GRPO has reached beyond reasoning to speech (Shivakumar et al., 2025) with supporting theory (Mroueh, 2025).

6 Conclusion

CTW reads token-level credit from a signal GRPO uses only at sequence level, the reward-induced partition of a group into correct and incorrect completions, exploiting it at token granularity at the cost of one frozen-scorer pass. As the cheap first stage of a precision-binding cascade, its teacher-free variant CTW-S lifts recall at fixed precision; and because it captures most of CTW-T's gain with no external model, it points toward collapsing the cascade into a single 4B detector and retiring the verifier. The natural next test is longer-horizon RLVR (code, instruction following), where the reward-induced partition signal may matter most.

Acknowledgments

We thank the anonymous reviewers and the area chair for their constructive feedback, which improved this paper.

7 Limitations

Our study leaves several questions open. The math experiments run on a deliberately small budget (160 steps, batch size 32), so we have not tested whether the gains from contrastive token weighting persist at larger step counts and batch sizes; scaling behavior is future work. We evaluate only two student sizes from a single model family (Qwen3-1.7B-Base and 4B-Base), so we do not know how the method interacts with other architectures and pretraining recipes. The MATH improvements may partly reflect an in-distribution effect, since the RL training distribution overlaps that benchmark more than it does GSM8K, and the smaller GSM8K margins fit that reading. Our index-aligned contrast assumes approximate positional correspondence between correct and incorrect completions, and although the clamp and normalization reduce the impact of misaligned positions, they do not fully remove position-misalignment noise. CTW consumes only the binary correct/incorrect partition, not the reward magnitude, so it is insensitive to reward-magnitude noise and is affected only by outright label flips, whose impact is further damped by averaging over the opposing subset; even so, we have not systematically characterized its sensitivity to group size or to partition quality under noisier reward models, and whether the mechanism extends to softer, continuous reward signals beyond binary-verifiable tasks remains an open question. Our evidence on reward quality is indirect (the cascade's audited 99.1% precision on $\sim$2,000 human-checked flags) rather than a systematic verifier-versus-human study on training rollouts. Finally, the deployment evidence is offline and scale-based (precision audit, coverage, and observational correlations between flags and both conversion and return rate across two storefronts); a randomized fix-versus-holdback experiment, the natural next step, would establish downstream business impact more directly than the metrics we report.

References

Tejas Alisyam, Prateek Sircar, and Deepak Gupta. 2026. When soft labels beat perfect rewards: Knowledge distillation vs. GRPO. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*. To appear.

Xinzhu Chen, Wei He, Huichuan Fan, Wenzhe Niu, Zhongxiang Sun, Xuanru Wang, Jiuchong Gao, Jinghua Hao, Renqing He, and Weijie Yu. 2026. Hidden states know where reasoning diverges: Credit assignment via span-level Wasserstein distance. *Preprint*, arXiv:2604.23318. ArXiv:2604.23318.

Kewei Cheng, Xian Li, Zhengyang Wang, Chenwei Zhang, Binxuan Huang, Yifan Ethan Xu, Xin Luna Dong, and Yizhou Sun. 2023. Tab-Cleaner: Weakly supervised tabular data cleaning via pre-training for E-commerce catalog. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL): Industry Track*, pages 172–185.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. 2025. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *International Conference on Machine Learning (ICML)*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, and 4 others. 2025. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*.

DeepSeek-AI. 2025. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638.

Chenhao Fang, Xiaohan Li, Zezhong Fan, Jianpeng Xu, Kaushiki Nag, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2024. LLM-ensemble: Optimal large language model ensemble method for E-commerce product attribute value extraction. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

Yiran Guo, Lijie Xu, Jie Liu, Dan Ye, and Shuang Qiu. 2025. Segment policy optimization: Effective segment-level credit assignment in RL for large language models. *Preprint*, arXiv:2505.23564. ArXiv:2505.23564.

Dan Hendrycks, Collin Burns, Saurav Kadavath, and 1 others. 2021. Measuring mathematical problem solving with the MATH dataset. *Advances in Neural Information Processing Systems*, 34.

Aniket Joshi, Cyrus DSouza, Sejal Jain, Jitenkumar Rana, and Promod Yenigalla. 2025. I-SEE: An instruction-tuned, SOP-enhanced quality evaluator for product content. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP): Industry Track*, pages 1379–1388.

Anant Khandelwal, Happy Mittal, Shreyas Sunil Kulkarni, and Deepak Gupta. 2023. Large scale generative multimodal attribute extraction for E-commerce attributes. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL): Industry Track*.

Yang Li, Gongle Xue, Yijia Guo, Yuheng Yuan, Liwen Hu, and Lei Ma. 2026a. CAST: Non-privileged clipped asymmetric self-teaching with advantage flipping for GRPO. *Preprint*, arXiv:2606.00172. ArXiv:2606.00172.

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

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Xingyu Lin, Yilin Wen, En Wang, Du Su, Wenbin Liu, Chenfu Bao, and Zhonghou Lv. 2025. Token-level policy optimization: Linking group-level rewards to token-level aggregation via Markov likelihood. *arXiv preprint arXiv:2510.09369*.

Aiwei Liu, Haoping Bai, Zhiyun Lu, Yanchao Sun, Xiang Kong, Simon Wang, Jiulong Shan, Albin Madappally Jose, Xiaojiang Liu, Lijie Wen, Philip S. Yu, and Meng Cao. 2025a. TIS-DPO: Token-level importance sampling for direct preference optimization with estimated weights. *International Conference on Learning Representations (ICLR)*.

Zheng Liu, Mengjie Liu, Siwei Wen, Mengzhang Cai, Bin Cui, Conghui He, and Wentao Zhang. 2025b. Heterogeneous adaptive policy optimization: Tailoring optimization to every token's nature. *arXiv preprint arXiv:2509.16591*.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025c. Understanding R1-Zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Youssef Mroueh. 2025. Reinforcement learning with verifiable rewards: GRPO's effective loss, dynamics, and success amplification. *Preprint*, arXiv:2503.06639. ArXiv:2503.06639.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, and 1 others. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Prashanth Gurunath Shivakumar, Yile Gu, Ankur Gandhe, and Ivan Bulyko. 2025. Group relative policy optimization for speech recognition. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*.

Wei Sun, Wen Yang, Pu Jian, Qianlong Du, Fuwei Cui, Shuo Ren, and Jiajun Zhang. 2025. KTAE: A model-free algorithm to key-tokens advantage estimation in mathematical reasoning. *arXiv preprint arXiv:2505.16826*.

Hongze Tan, Zihan Wang, Jianfei Pan, Jinghao Lin, Hao Wang, Yifan Wu, Tao Chen, Zhihang Zheng, Zhihao Tang, and Haihua Yang. 2025. GTPO and GRPO-S: Token and sequence-level reward shaping with policy entropy. *arXiv preprint arXiv:2508.04349*.

Akshay Verma, Swapnil Gupta, Deepak Gupta, Prateek Sircar, and Siddharth Pillai. 2026. SELENE: Selective and evidence-weighted LLM debating for efficient and reliable reasoning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 95–104.

Vinay Kumar Verma, Shreyas Sunil Kulkarni, Happy Mittal, and Deepak Gupta. 2025. MoEMoE: Question guided dense and scalable sparse mixture-of-expert for multi-source multi-modal answering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics (NAACL)*.

Peiyi Wang, Lei Li, Zhihong Shao, and 1 others. 2023. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*.

Shenzhi Wang, Le Yu, Chang Gao, and 1 others. 2025a. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. *Preprint*, arXiv:2506.01939. ArXiv:2506.01939.

Yining Wang, Jinman Zhao, Chuangxin Zhao, Shuhao Guan, Gerald Penn, and Shinan Liu. 2025b. λ-GRPO: Unifying the GRPO frameworks with learnable token preferences. *arXiv preprint arXiv:2510.06870*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, and 1 others. 2025. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. 2024. Token-level direct preference optimization. *International Conference on Machine Learning (ICML)*.

Songming Zhang, Xue Zhang, Tong Zhang, Bojie Hu, Yufeng Chen, and Jinan Xu. 2025. AlignDistil: Token-level language model alignment as adaptive policy distillation. *Association for Computational Linguistics (ACL)*.

Yuzhong Zhao, Yue Liu, Junpeng Liu, and 1 others. 2025. Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*.

Han Zhong, Zikang Shan, Guhao Feng, Wei Xiong, and 1 others. 2025. DPO meets PPO: Reinforced token optimization for RLHF. *International Conference on Machine Learning (ICML)*.

A Method Details

Algorithm 1 gives the full CTW-weighted GRPO procedure.

Algorithm 1 CTW-Weighted GRPO

Require: Policy π_θ, frozen scorer π_T (teacher for CTW-T; student copy for CTW-S), prompts $\{x\}$, group size G, α, $c_{\min}$

for each prompt x **do**
  Sample G completions $\{y_1, \ldots, y_G\} \sim \pi_\theta(\cdot \mid x)$
  Compute rewards $\{r_1, \ldots, r_G\}$ and group-relative advantages $\{A_1, \ldots, A_G\}$
  Partition: $\mathcal{C} = \{i : A_i > 0\}$, $\mathcal{I} = \{i : A_i \leq 0\}$
  Compute scorer log-probs $\ell_{i,t} = \log \pi_T(y_{i,t} \mid x, y_{i,<t})$ for all i, t
  if $|\mathcal{C}| > 0$ and $|\mathcal{I}| > 0$ **then**
    Compute contrastive signal $s_{i,t}$ (Eq. 3)
    Normalize: $\bar{s}_{i,t} = s_{i,t} / (\text{mean}_t(s_{i,t}) + \epsilon)$
    Compute weights: $w_{i,t} = \text{clamp}(1 + \alpha(\bar{s}_{i,t} - 1), c_{\min})$
  else
    Set $w_{i,t} = 1$ (uniform fallback; recovers vanilla GRPO for this group)
  end if
  Update θ using the CTW-weighted loss (Section 2)
end for

Table 3 lists the full hyperparameter settings for both setups, which share every value and differ only in the loss weighting. Figure 3 visualizes the resulting CTW-T weights on a sample problem.

B Production System: Supporting Material

Table 4 shows representative title–bullet inconsistencies flagged by the deployed detector, sampled from high-traffic listings (>700K page views over the trailing 90 days) and spanning several product categories. In each case the title and bullets refer to semantically different products or attributes, which a simple string-match heuristic would miss.

Table 5 breaks the flag rate down by product line: apparel and soft goods carry both the highest flag rate and the most affected traffic, consistent with free-text, attribute-rich listings (material, fit, size) being the most error-prone, while consumables with simpler descriptions are flagged least often.

Table 3: Full hyperparameter settings for both experimental setups.

Hyperparameter	**Product inconsistency**	**Math reasoning**
Student model	Qwen3-4B-Instruct (SFT)	Qwen3-1.7B-Base, Qwen3-4B-Base
Reward model	DeepSeek R1 (verifier)	Verifiable answer match
Variant(s)	CTW-S (deployed)	CTW-T and CTW-S
Scorer π_T	Frozen student copy (CTW-S)	Qwen3-8B-Base (CTW-T)
LoRA rank r	32	— (full fine-tune)
LoRA α	16	—
LoRA dropout	0.05	—
Learning rate	3×10^{-5}	3×10^{-5}
Training steps	160	160
Group size G	8	8
Effective batch size	32	32
Max completion length	2,048	2,048
CTW α	0.1	0.1
CTW c_{min}	0.5	0.5
CTW ϵ (mean-norm)	10^{-8}	10^{-8}
Optimizer	AdamW (torch fused)	AdamW (torch fused)
Optimizer β_1	0.9	0.9
Optimizer β_2	0.999	0.999
Optimizer ϵ	10^{-8}	10^{-8}
Prompting	—	2-shot Qwen-boxed

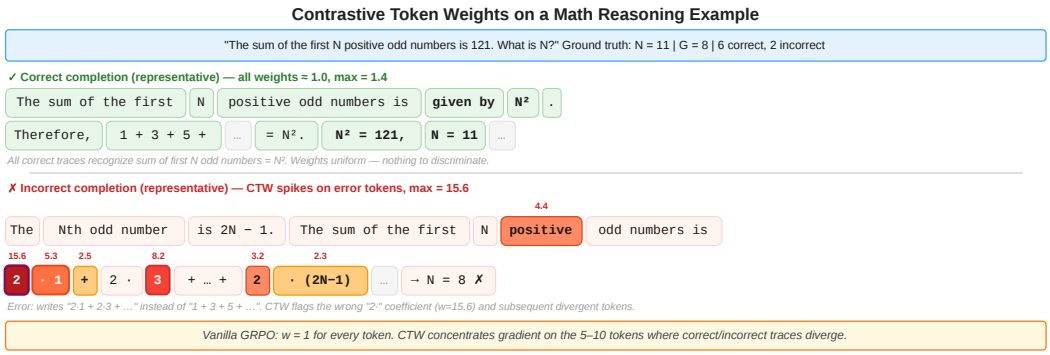


Figure 3: CTW-T weights on a MATH problem (Qwen3-1.7B student, 8B teacher). Both subsets receive a contrastive signal (Eq. 3); the weights concentrate on a few tokens where the scorer most sharply separates correct from incorrect completions, while most positions stay near the uniform weight of 1.

Table 4: Representative inconsistency examples flagged by the deployed detector (high-traffic listings, >700K page views over the trailing 90 days).

Cause of inconsistency	Title and bullet excerpts
Storage capacity mismatch: title advertises a phone with 128GB storage, but the memory bullet specifies 512GB.	**Title:** "Vivo V60 5G (Mist Gray, 8GB RAM, *128GB* Storage)…" **Bullet 3:** "[MEMORY & SIM]: 12GB RAM | *512GB* Internal Memory | LPDDR4X…"
Fabric composition mismatch: title labels the polo as "Cotton", but the material bullet specifies a blended fabric.	**Title:** "Allen Solly Men's Plain Regular Fit *Cotton* Polo…" **Bullet 3:** "Material: *60% Cotton and 40% Polyester*."
Energy rating mismatch: title claims a 3-star energy rating, but the rating bullet specifies 2-star.	**Title:** "Haier SmartChoice 596L *3Star* 2-Door Side by Side Frost Free Refrigerator…" **Bullet 3:** "Energy Rating: *2 Star* Energy Efficiency | Annual Energy Consumption: 528 kWh/year."
Battery-life mismatch: title advertises up to 35-hour battery life, but the battery bullet claims up to 50 hours.	**Title:** "Sony WH-CH720N…Up to *35Hrs* Battery…" **Bullet 2:** "Up to *50-hour* battery life with quick charging…"

Table 5: Flagged inconsistencies by product line. Flag rate is inconsistent listings as a fraction of listings in that line.

Product line	Listings	Flagged	Flag rate
Apparel & soft goods	16.4M	1.48M	9.0%
Hardlines	13.1M	840K	6.4%
Consumer electronics	4.67M	358K	7.7%
Consumables	1.78M	64K	3.6%

Figure 4 shows the conversion penalty by traffic decile on the primary storefront (Section 3).

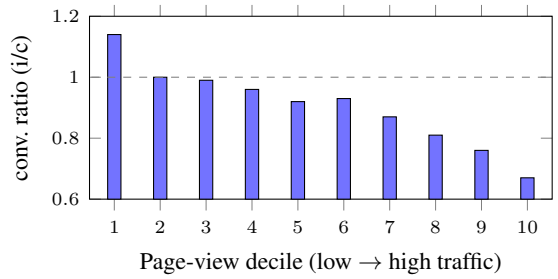


Figure 4: Inconsistent-to-consistent conversion ratio by traffic decile (primary storefront). Near parity at low (noise-dominated) traffic; the penalty deepens monotonically to ∼0.67 in the highest-traffic decile. This is observational, not a causal fix effect.

C Math Study: Training Curves

Figure 5 plots held-out MATH accuracy over training for the controlled study (Section 4).

D Additional Ablations

Robustness to CTW hyperparameters. A coarse 3×3 sweep over $\alpha \in \{0.05, 0.1, 0.3\}$ and $c_{\text{min}} \in \{0.3, 0.5, 0.7\}$ at 1.7B (single seed, 160 steps) shows that all nine configurations beat Vanilla GRPO on aggregate, so CTW's gains are robust to the hyperparameter choice rather than tied to a narrow optimum.

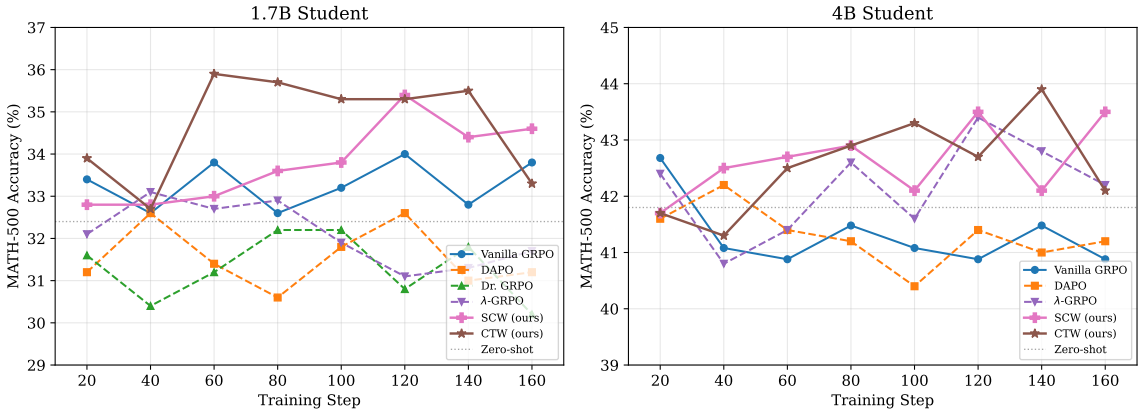


Figure 5: MATH accuracy (%) over training steps for 1.7B (left) and 4B (right) students (seed 42). CTW-T and CTW-S separate from baselines by step 60 and stabilize by step 120–160; non-contrastive baselines remain near the zero-shot level (dashed gray) throughout.