# CONTEXTUAL CREDIT DIFFUSION FOR LLM POST-TRAINING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

LLM post-training has long relied on a single objective at each position: optimizing a weighted log-likelihood of a single reference token in the vocabulary. This reference token is either the sampled action in RL methods such as PPO and GRPO, or the ground-truth label in SFT. The loss can be written in the unified form $\mathcal{L} = -\mathbb{E}_t \left[ w_t \cdot \log \pi_\theta(a_t^{\text{ref}} \mid s_t) \right]$, where $w_t$ is an advantage in RL methods or a constant in SFT. However, solving a reasoning problem often admits multiple valid paths that all lead to the correct answer, yet the forced-reference objective locks the model onto a single path at each step and penalizes alternative tokens in the vocabulary that could also lead to a valid answer. We propose **Contextual Credit Diffusion (CCD)**. CCD diffuses the reference token's credit across the vocabulary according to their contextual interchangeability, measured by the logit difference in the model's next-token distribution under the current context, replacing the single-reference constraint with a dense reward. **This offers a potential solution to both the credit assignment problem in RL and the one-hot label problem in SFT.** CCD consistently outperforms RL and SFT baselines on six reasoning benchmarks, achieving significant gains. Furthermore, a token-replacement experiment shows that replacing the reference token with candidate tokens whose diffused credit is high yields the same final answer in most cases, confirming that the reward captures contextual interchangeability and identifies alternative valid paths.

## 1 INTRODUCTION

LLM's post-training has historically followed two main paradigms. Reinforcement learning methods like PPO (Schulman et al., 2017) and GRPO (Guo et al., 2025; Shao et al., 2024) sample an action from the model's own policy and optimize its advantage-weighted log-likelihood, where a positive advantage increases the action's probability and a negative advantage decreases it. Supervised fine-tuning (SFT) forces the model to maximize the log-likelihood of a ground-truth label token at each position. Despite their apparent differences, these paradigms share a common mathematical structure: both optimize the log-probability of a single forced reference token—the sampled action in RL, the label in SFT. Both objectives provide direct training credit to a single reference token. However, when an LLM predicts the next token, it outputs a probability over the entire vocabulary. Although other tokens in the vocabulary are affected through softmax normalization, the training objectives never explicitly distinguish them.

For reasoning tasks, this single-reference rigidity may become harmful. Solving a math problem or tracing a logical chain often admits multiple valid paths that all lead to the correct answer. A model might write "x = 3", "the answer is 3", or "the value of x is 3"—all semantically equivalent. Yet the forced-reference objectives penalize all other non-reference tokens in the vocabulary, while some of them could also lead to a valid answer.

Recent work has attempted to address this limitation in SFT from policy gradient perspectives. Dynamic Fine-Tuning (DFT) (Wu et al., 2025) modifies the standard cross-entropy objective by multiplying it with a correction term derived from the model's own output probability. One-Token Rollout (OTR) (Ming et al., 2025) instead rolls out candidate tokens from the current policy and directly optimizes a token-level RL objective. However, although they assign rewards to non-reference tokens in the vocabulary, they treat every token identically: DFT gives 0 to all non-reference tokens, and OTR gives them a negative constant. Such uniform treatment is clearly flawed: non-reference



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

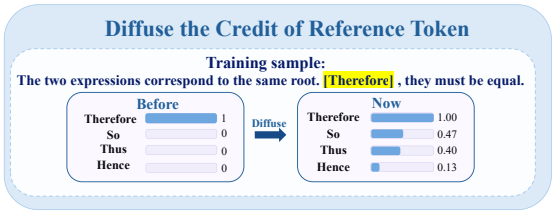



Figure 1: From single-token credit to context-aware dense rewards. CCD diffuses the credit of the reference token to other candidates.

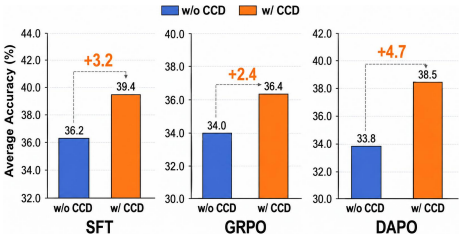



Figure 2: Average accuracy with and without CCD on Qwen3-8B-Base under SFT, GRPO, and DAPO.

tokens should not be valued equally. If the reference token is "0", then "zero", which is semantically related, should receive a higher reward than an unrelated token such as "apple". Yet existing methods fail to distinguish them, discarding the fine-grained semantic information that reasoning tasks depend on.

In the RL setting, a prominent line of work addresses this problem through On-Policy Distillation (OPD) (Agarwal et al., 2024; Song & Zheng, 2026). OPD leverages a stronger teacher model to provide dense, token-level supervision on data generated by the student model's own policy. While effective, this approach relies critically on an additional teacher model, which significantly increases training overhead and deployment complexity.

We propose **Contextual Credit Diffusion (CCD)**. Instead of concentrating the training signal on a single reference token, CCD diffuses its credit to nearby candidate tokens. The diffusion is guided by the model's own predictive structure under the current context. The key insight is that the last-layer hidden state at each step encodes the model's understanding of the entire context. A token's output logit is determined solely by the product of this hidden state and the corresponding column of the LLM head, which can be viewed as the projection of that token's vector in the LLM head onto the direction of context in a high-dimensional semantic space. Therefore, the product of the hidden state with the difference between two token vectors in the LLM head directly measures their predictive proximity under the current context, which is exactly the logit difference. We define a **CCD coefficient** that quantifies how interchangeable a candidate token is with the reference token under the current context based on this logit gap. Candidates with smaller gaps are closer to the reference and receive more diffused credit, while distant candidates receive less. In this way, CCD turns single-reference supervision into context-dependent local credit diffusion. The reference token retains the strongest signal, while part of its credit is smoothly propagated to nearby alternatives.

Importantly, CCD operates at the level of next-token prediction—the fundamental working principle of all LLMs—by assigning dense rewards across the vocabulary at each time step. This makes CCD independent of any specific training paradigm: it can be plugged into SFT, GRPO, PPO, or any method that optimizes token-level decisions. Moreover, CCD introduces **negligible training overhead**, which we will detail in Section 5.

To directly validate that CCD's reward captures genuine contextual interchangeability, we sample 100,000 candidate tokens with CCD coefficient above $0.1$ and replace the reference token at their positions. We then compare the final outcome generated from the reference token and the candidate token. Remarkably, when the CCD coefficient is above $0.1$, the two final answers agree in over $80\%$ of cases; when the coefficient exceeds $0.9$, the agreement rate reaches $100\%$. These high agreement rates provide direct empirical evidence that CCD's reward effectively identifies alternative tokens that lead to the same outcome, confirming that it rewards different paths that converge to the same answer. We evaluate CCD on six reasoning benchmarks. Under both RL and SFT settings, CCD consistently improves over baselines.

Our contributions can be summarized as follows:

- We propose **Contextual Credit Diffusion (CCD)**, a novel training paradigm that diffuses the credit of a single reference token to other tokens in the vocabulary based on their contextual predictive proximity to the reference. This dense reward mechanism offers a potential

solution to both the credit assignment problem in RL and the label-fitting problem in SFT, without requiring any additional teacher model.

- Through a replacement experiment on 100,000 candidate tokens, we demonstrate that tokens with high CCD coefficient lead to the same final outcomes as the reference token in most cases, providing evidence that CCD's reward captures alternative valid paths.
- We evaluate CCD on six reasoning benchmarks under both RL and SFT pipelines. CCD shows consistent improvements over baselines, verifying its effectiveness in both RL and SFT.

## 2 PRELIMINARIES

We first formalize the training objectives for RL and SFT methods. We then unify them under a common policy-gradient view, which lays the foundation for our proposed method.

### 2.1 REINFORCEMENT LEARNING VIA POLICY OPTIMIZATION METHODS

Policy optimization methods such as PPO and GRPO optimize the model toward maximizing expected rewards from verifiable outcomes. These methods share a common structure: they sample actions from the current policy, compute an advantage $A_t$ for each token position, and optimize the following objective:

$$\mathcal{L}_{\text{RL}} = -\frac{1}{T}\sum_{t=1}^{T} A_t \cdot \log \pi_\theta(a_t \mid s_t),$$

where $\pi_\theta$ is the model's policy distribution over the vocabulary, $a_t$ is the token sampled by the model at time step $t$, and $s_t$ denotes the context history up to time $t$.

### 2.2 SUPERVISED FINE-TUNING

Supervised fine-tuning (SFT) trains a language model to predict the next token given a context. Given a sequence of tokens of length $T$, the model maximizes the log-likelihood of each ground-truth token conditioned on previous tokens. The loss for a single sequence is:

$$\mathcal{L}_{\text{SFT}} = -\frac{1}{T}\sum_{t=1}^{T} \log \pi_\theta(y_t \mid s_t),$$

where $y_t$ is the ground-truth label token at position $t$.

### 2.3 A UNIFIED VIEW: FORCED-REFERENCE POLICY GRADIENT

Despite their apparent differences, RL and SFT share a common mathematical structure. Observe that both objectives can be written as:

$$\mathcal{L}_{\text{RL}} = -\frac{1}{T}\sum_{t=1}^{T} \underbrace{A_t}_{w_t} \cdot \log \pi_\theta(\underbrace{a_t}_{a_t^{\text{ref}}} \mid s_t).$$

$$\mathcal{L}_{\text{SFT}} = -\frac{1}{T}\sum_{t=1}^{T} \underbrace{1}_{w_t} \cdot \log \pi_\theta(\underbrace{y_t}_{a_t^{\text{ref}}} \mid s_t),$$

In both cases, at each token position $t$, the objective optimizes the log-probability of a single reference token $a_t^{\text{ref}}$ (the sampled action in RL, the label in SFT), scaled by a weight $w_t$ (constant 1 in SFT, advantage $A_t$ in policy optimization). This yields a unified formulation:

$$\mathcal{L} = -\frac{1}{T}\sum_{t=1}^{T} w_t \cdot \log \pi_\theta(a_t^{\text{ref}} \mid s_t).$$

This **single-reference** paradigm forces the model to optimize the log-probability of a single reference token at each position, increasing it under a positive advantage and decreasing it under a negative one. While other tokens obtain gradients through softmax normalization, they receive no explicit optimization signal, even though the model naturally assigns them non-zero probabilities.

## 3 NEXT-TOKEN DECISION FRAMEWORK

This paper reframes next-token prediction as a one-step decision-making problem. At each time step $t$, the model observes a state $s_t$ (the context history) and selects an action from its policy distribution $\pi_\theta(\cdot \mid s_t)$ over the vocabulary $\mathcal{V}$. We define a reward function $r(a, s_t)$ that assigns a scalar reward to each candidate action $a \in \mathcal{V}$.

The training objective then maximizes the expected reward under the policy:

$$\mathcal{J}(\theta) = \frac{1}{T} \sum_{t=1}^{T} \sum_{a \in \mathcal{V}} \pi_\theta(a \mid s_t) \cdot r(a, s_t).$$

To maximize $\mathcal{J}(\theta)$ via gradient ascent, we compute its gradient:

$$\nabla_\theta \mathcal{J}(\theta) = \frac{1}{T} \sum_{t=1}^{T} \sum_{a \in \mathcal{V}} r(a, s_t) \cdot \nabla_\theta \pi_\theta(a \mid s_t).$$

Applying the log-derivative trick $\nabla_\theta \pi = \pi \cdot \nabla_\theta \log \pi$, we obtain:

$$\nabla_\theta \mathcal{J}(\theta) = \frac{1}{T} \sum_{t=1}^{T} \sum_{a \in \mathcal{V}} \pi_\theta(a \mid s_t) \cdot r(a, s_t) \cdot \nabla_\theta \log \pi_\theta(a \mid s_t).$$

In this expression, $\pi_\theta(a \mid s_t)$ serves as a weighting coefficient and should be treated as a constant with respect to the gradient. Therefore, we can define a surrogate loss:

$$\mathcal{L}(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{a \in \mathcal{V}} \text{sg}[\pi_\theta(a \mid s_t)] \cdot r(a, s_t) \cdot \log \pi_\theta(a \mid s_t),$$

where $\text{sg}[\cdot]$ denotes stop-gradient. Minimizing $\mathcal{L}(\theta)$ via gradient descent yields the same gradient as maximizing $\mathcal{J}(\theta)$.

Within this framework, existing methods can be seen as specific choices of the reward function $r(a, s_t)$:

- **SFT and RL:** From the next-token decision perspective, these two paradigms can be understood from two equivalent angles:
  - **Reward view:** The reward for the reference token is $1/\pi_\theta(a_t \mid s_t)$ in SFT and $A_t/\pi_\theta(a_t \mid s_t)$ in RL, while all other tokens receive 0.
  - **Sampling approximation view:** The expectation over the full vocabulary is approximated by a single sampled token. SFT forces the label token, whose reward is always 1. RL samples an action from the current policy, whose reward is the advantage $A_t$.
- **DFT** (Wu et al., 2025): $r(a, s_t) = 1$ if $a = y_t$, and 0 otherwise.
- **OTR** (Ming et al., 2025): $r(a, s_t) = 1$ if $a = y_t$, and a negative constant otherwise.
- **OPD** (Agarwal et al., 2024): $r(a, s_t) = \log \frac{\pi_{\text{teacher}}(a|s_t)}{\pi_\theta(a|s_t)}$, where $\pi_{\text{teacher}}$ is the teacher model's probability.

OPD distinguishes each token based on the KL divergence between the teacher and student distributions, but it requires an additional teacher model. Other methods share the same limitation: they assign identical rewards to all non-reference tokens. What's more, DFT and OTR both operate on SFT, and to our knowledge, no work has improved RL in this regard. Our proposed method, CCD, replaces this uniform reward with a dense, context-aware reward that reflects each candidate's contextual predictive proximity to the reference. CCD can be plugged into both SFT and RL.

# 4 METHOD

## 4.1 HOW TO DESIGN REWARDS

The one-token decision framework in Section 3 reduces the training objective to a choice of the reward function $r(a, s_t)$. Designing such a reward function is far from trivial, as it requires defining a meaningful scalar for each token at each position that reflects the model's current context and semantic objective. We answer this question through two insights that guide our reward design.

### 4.1.1 INSIGHTS OF REWARD DESIGN

**Insight 1: Reward Depends on Context.** Language models predict the next token based on the preceding context. The same token can carry entirely different meanings in different contexts. The token "3" in a math problem means a numeric value; in an address, it means a house number. Therefore, the reward must be context-aware.

**Insight 2: Reward Relates to the Reference Token.** In both RL and SFT, the reference token provides the primary training signal. Therefore, the reward must reflect the relationship between each candidate and the reference token.

These two insights together point to a natural design: the reward of a candidate token should be determined by its relationship to the reference token under the current context.

**Contextual Interchangeability.** Given the same preceding context $s_t$, we say that two tokens are *contextually interchangeable* if choosing either token at position $t$ can lead to subsequent continuations that preserve the same underlying semantics or downstream outcome. Importantly, contextual interchangeability is not an intrinsic property of the token pair, but depends on the context in which the tokens are considered.

Our goal is therefore to assign higher rewards to candidate tokens that are more interchangeable with the reference token under the current context. Directly measuring such interchangeability, however, would require rolling out the continuation from each candidate token and evaluating its downstream effect, which is prohibitively expensive over the vocabulary. We therefore seek an efficient proxy that can be obtained directly from the current model.

**The key to our method is to leverage the LLM's own powerful language modeling capability.** We use the model's context-conditioned predictive structure to estimate how interchangeable a candidate token is with the reference token, and translate this estimate into a dense reward.

### 4.1.2 CONTEXTUAL PREDICTIVE DISTANCE IN LANGUAGE MODELS

Here we explain why the difference between the logits of two tokens is sufficient to capture how interchangeable they are under the current context.

Language models encode the context into the last-layer hidden state $h_t \in \mathbb{R}^d$ at each step $t$. Its norm $\|h_t\|$ is stabilized by the preceding layer normalization, so $h_t$ can be viewed as a context-defined direction in the high-dimensional space. The model also maintains a projection matrix $W \in \mathbb{R}^{d\times|\mathcal{V}|}$, which is the language model head. Each column $W_a$ corresponds to a token $a$ in the vocabulary. The probability over the entire vocabulary is obtained by applying softmax to the logits, where the logit for token $a$ is

$$\text{logit}(a) = h_t \cdot W_a.$$

This can be viewed as the projection of the token's corresponding vector in the projection matrix onto the context direction. The longer this projection, the higher the token's logit and thus its probability. Through pretraining, the model learns the relationship between each token's vector and the various context directions it encounters, so that the same token can exhibit drastically different projection lengths under different contexts.

Thus, if we consider the projection of the difference between two token vectors onto the context direction, we obtain:

$$|h_t \cdot (W_a - W_b)| = |\text{logit}(a) - \text{logit}(b)|.$$

This quantity measures the model's contextual predictive discrepancy between the two tokens: how differently the model scores them under the current context. A small discrepancy indicates that the model treats the two tokens as near-interchangeable choices, while a large discrepancy indicates a clear preference. Since $\|h_t\|$ is bounded, this discrepancy reflects directional alignment rather than vector magnitude.

#### 4.1.3 Reward as a Contextual Predictive Function

The contextual predictive discrepancy between a candidate token $a$ and the reference token $a_t^{\text{ref}}$ is

$$d(a, a_t^{\text{ref}}) = |h_t \cdot (W_a - W_{a_t^{\text{ref}}})| = |\text{logit}(a) - \text{logit}(a_t^{\text{ref}})|.$$

We convert this discrepancy into a proximity coefficient, which we term the **CCD coefficient**, via exponential decay:

$$\phi(a, s_t) = \exp\left(-\frac{|h_t \cdot (W_a - W_{a_t^{\text{ref}}})|}{\tau}\right),$$

where $\tau$ is a scaling factor that controls the smoothness of the coefficient distribution. This coefficient measures how close token $a$ is to the reference token under the current context. When $a = a_t^{\text{ref}}$, the discrepancy is $0$ and the coefficient is $1$; as $a$ moves farther away, the coefficient decays toward $0$. This coefficient is computed with stop-gradient, as it serves only as a weighting signal rather than a differentiable objective.

We then diffuse the credit of the reference token across the vocabulary by scaling this coefficient with $w_t$:

$$r_{\text{CCD}}(a, s_t) = w_t \cdot \exp\left(-\frac{|h_t \cdot (W_a - W_{a_t^{\text{ref}}})|}{\tau}\right) = \begin{cases} A_t \cdot \exp\left(-\dfrac{|h_t \cdot (W_a - W_{a_t^{\text{ref}}})|}{\tau}\right), & \text{for RL,} \\ 1 \cdot \exp\left(-\dfrac{|h_t \cdot (W_a - W_{a_t^{\text{ref}}})|}{\tau}\right), & \text{for SFT,} \end{cases}$$

where $A_t$ is the advantage in RL. Instead of concentrating all credit on a single reference token, this reward spreads it over tokens that the model considers near-interchangeable with the reference, with the magnitude of each token's share determined by its contextual predictive proximity.

### 4.2 Contextual Credit Diffusion

As discussed in Section 3, the training objective of CCD is to maximize the expected reward under the current policy:

$$\mathcal{J} = \frac{1}{T} \sum_{t=1}^{T} \sum_{a \in \mathcal{V}} \pi_\theta(a \mid x_{<t}) \cdot r_{\text{CCD}}(a, x_{<t}).$$

In practice, we restrict the summation to the top-$k$ tokens at each position:

$$\mathcal{J}^{\text{top-}k} = \frac{1}{T} \sum_{t=1}^{T} \sum_{a \in \mathcal{T}_t} \pi_\theta(a \mid x_{<t}) \cdot r_{\text{CCD}}(a, x_{<t}),$$

where $\mathcal{T}_t = \text{TopK}(\pi_\theta(\cdot \mid x_{<t}))$. This is justified for two reasons: first, the manifold of natural language is low-dimensional, so at any given position only a small subset of the vocabulary forms a meaningful continuation; second, it significantly reduces memory usage during training. Note that we always include $a_t^{\text{ref}}$ in $\mathcal{T}_t$, even if it is not among the top-$k$ tokens, so that the primary supervision signal is never removed. Ablation studies on top-$k$ are provided in Appendix B.

For RL, the objective and the corresponding surrogate loss are

$$\mathcal{J}_{\text{RL}} = \frac{1}{T} \sum_{t=1}^{T} A_t \sum_{a \in \mathcal{T}_t} \pi_\theta(a \mid x_{<t}) \cdot \exp\left(-\frac{|h_t \cdot (W_a - W_{a_t})|}{\tau}\right),$$

$$\mathcal{L}_{\text{RL}} = -\frac{1}{T}\sum_{t=1}^{T} A_t \sum_{a\in\mathcal{T}_t} \text{sg}[\pi_\theta(a \mid x_{<t})] \cdot \text{sg}\left[\exp\left(-\frac{|h_t \cdot (W_a - W_{a_t})|}{\tau}\right)\right] \cdot \log \pi_\theta(a \mid x_{<t}).$$

For SFT, the objective and the corresponding surrogate loss are

$$\mathcal{J}_{\text{SFT}} = \frac{1}{T}\sum_{t=1}^{T} \sum_{a\in\mathcal{T}_t} \pi_\theta(a \mid x_{<t}) \cdot \exp\left(-\frac{|h_t \cdot (W_a - W_{y_t})|}{\tau}\right),$$

$$\mathcal{L}_{\text{SFT}} = -\frac{1}{T}\sum_{t=1}^{T} \sum_{a\in\mathcal{T}_t} \text{sg}[\pi_\theta(a \mid x_{<t})] \cdot \text{sg}\left[\exp\left(-\frac{|h_t \cdot (W_a - W_{y_t})|}{\tau}\right)\right] \cdot \log \pi_\theta(a \mid x_{<t}).$$

where $\text{sg}[\cdot]$ denotes stop-gradient. We use $k = 10$ in all experiments.

# 5 Experiments

## 5.1 Experiment Setup

**Models.** For RL experiments, we use Qwen3-1.7B-Base, Qwen3-4B-Base and Qwen3-8B-Base. For SFT experiments, we use Qwen3-4B-Base and Qwen3-8B-Base.

**Training Datasets.** For RL experiments, we use DAPO-MATH-17K (Yu et al., 2025), a dataset of mathematical problems with verifiable answers. For SFT experiments, we use the Int (Yang et al., 2026a) dataset, which contains over 4,500 high-quality mathematical competition problems with step-by-step solutions.

**Baselines.** For RL experiments, we compare against standard GRPO and DAPO (Yu et al., 2025). Since CCD is a plug-in reward mechanism that operates at the token level, we evaluate it as GRPO+CCD and DAPO+CCD to demonstrate its compatibility with existing policy optimization methods. For SFT experiments, we compare against SFT, DFT (Wu et al., 2025), EAFT (Diao et al., 2026), and Label Smoothing (Szegedy et al., 2016).

**Evaluation Benchmarks.** We evaluate on six mathematical reasoning benchmarks: AIME24 and AIME25, AMC, MATH500, Minerva, and OlympiadBench.

**Training Details.** For RL experiments, we use the verl (Sheng et al., 2025) framework. For GRPO, we train for 3 epochs with a training batch size of 64 (660 steps), sampling 8 responses per prompt and computing group-normalized advantages. For DAPO, we train for 10 epochs with a generation batch size of 256 and a training batch size of 64 (550 steps). For SFT experiments, we train all models for 3 epochs. For CCD, we set $\tau = 5.0$ for RL experiments and $\tau = 1.0$ for SFT experiments, with $k = 10$ for the top-$k$ truncation in all experiments. Ablation studies on these hyperparameters are provided in Appendix B.

## 5.2 Main Results

Tables 1 and 2 show that CCD consistently improves both RL and SFT across all model scales. In RL, CCD raises GRPO and DAPO by 2.4–4.7 points on average. In SFT, CCD outperforms standard SFT by 2.9–3.2 points. These results demonstrate that CCD is a general and effective token-level training objective for reasoning tasks, regardless of whether the reference token comes from a label or a sampled action. Training dynamics are provided in Appendix B.

## 5.3 Replacement experiment: Validation of Contextual Interchangeability

To directly validate that CCD's proximity coefficient captures genuine contextual interchangeability, we design a substitution experiment. The core question is: *if we replace a reference token with another token that receives a high CCD coefficient, does the model still arrive at the same final answer?*

Table 1: Performance on mathematical reasoning benchmarks (avg@16, %).

| Method | AIME24 | AIME25 | AMC | MATH500 | Minerva | Olympiad | Avg |
|---|---|---|---|---|---|---|---|
| **Qwen3-1.7B-Base** | | | | | | | |
| Initial Model | 4.1 | 1.7 | 23.2 | 48.9 | 8.9 | 17.1 | 17.3 |
| GRPO | 5.0 | 2.6 | 25.7 | 58.4 | 11.3 | 20.8 | 20.6 |
| **GRPO+CCD** | **14.0** | **4.2** | **28.9** | **64.7** | **14.6** | **23.5** | **25.0** |
| DAPO | 6.7 | 2.3 | **28.1** | 57.5 | 10.6 | 19.8 | 20.8 |
| **DAPO+CCD** | **12.1** | **4.8** | 27.3 | **68.3** | **13.1** | **21.9** | **24.6** |
| **Qwen3-4B-Base** | | | | | | | |
| Initial Model | 9.3 | 6.1 | 36.6 | 60.6 | 14.8 | 27.1 | 25.8 |
| GRPO | 10.6 | 13.1 | 40.3 | 74.2 | 17.2 | 30.0 | 30.9 |
| **GRPO+CCD** | **11.4** | **15.2** | **45.3** | **75.2** | **21.1** | **34.4** | **33.8** |
| DAPO | **9.6** | 8.4 | 41.7 | 74.4 | 16.8 | 31.5 | 30.4 |
| **DAPO+CCD** | 7.3 | **12.7** | **46.5** | **75.4** | **20.0** | **34.8** | **32.8** |
| **Qwen3-8B-Base** | | | | | | | |
| Initial Model | 10.8 | 8.5 | 37.6 | 61.1 | 15.6 | 27.7 | 26.9 |
| GRPO | 12.5 | 14.1 | 48.0 | 74.5 | 20.4 | 34.5 | 34.0 |
| **GRPO+CCD** | **18.5** | **17.1** | **48.5** | **76.4** | **22.0** | **36.1** | **36.4** |
| DAPO | 8.3 | 12.5 | 50.6 | 76.7 | 16.8 | **38.0** | 33.8 |
| **DAPO+CCD** | **13.8** | **22.3** | **52.3** | **78.7** | **26.2** | 37.9 | **38.5** |

Table 2: SFT results on mathematical reasoning benchmarks (avg@16, %).

| Model | Method | AIME24 | AIME25 | AMC | MATH500 | Minerva | Olympiad | Avg |
|---|---|---|---|---|---|---|---|---|
| **Qwen3-4B-Base** | Initial Model | 9.3 | 6.1 | 36.6 | 60.6 | 14.8 | 27.1 | 25.8 |
| | SFT | 15.0 | 12.7 | 43.7 | 71.9 | 22.7 | **36.5** | 33.8 |
| | DFT | **15.6** | 13.1 | 44.5 | 76.7 | 29.4 | 34.1 | 35.6 |
| | EAFT | 12.7 | 16.0 | 45.6 | 70.0 | 22.6 | 33.5 | 33.4 |
| | Label Smoothing | 5.8 | 8.5 | 33.4 | 53.5 | 16.2 | 24.7 | 23.7 |
| | **CCD (ours)** | 14.0 | **16.8** | **46.7** | **76.9** | **30.3** | 35.6 | **36.7** |
| **Qwen3-8B-Base** | Initial Model | 10.8 | 8.5 | 37.6 | 61.1 | 15.6 | 27.7 | 26.9 |
| | SFT | 17.5 | **17.1** | 50.1 | 70.6 | 24.6 | 37.1 | 36.2 |
| | DFT | 15.2 | 14.0 | 50.8 | 80.1 | 27.8 | 39.5 | 37.9 |
| | EAFT | **20.4** | 16.2 | 50.2 | 66.2 | 21.8 | 35.6 | 35.1 |
| | Label Smoothing | 13.1 | 12.9 | 41.2 | 58.5 | 17.4 | 30.4 | 28.9 |
| | **CCD (ours)** | 18.1 | **17.1** | **51.6** | **80.6** | **28.8** | **40.0** | **39.4** |

**Experimental Design.** We sample 100,000 candidate tokens from the training data whose proximity coefficient exceeds $0.1$ and which are not the reference token. For each candidate token $a^{\text{cand}}$, we identify its position $t$ and the corresponding reference token $a^{\text{ref}}$, and compute the CCD proximity coefficient relative to $a^{\text{ref}}$ under the current context as defined in Section 4:

$$\phi(a^{\text{cand}}, x_{<t}) = \exp\left(-\frac{|\text{logit}(a^{\text{cand}}) - \text{logit}(a^{\text{ref}})|}{\tau}\right).$$

For each sampled candidate, we perform greedy decoding at its corresponding position under two conditions: (1) using the reference token $a^{\text{ref}}$, and (2) using the candidate token $a^{\text{cand}}$ at that position. We then compare whether the final answers generated under these two conditions are identical.

**Results.** The results are striking. For candidate tokens with proximity coefficient above $0.1$, the two continuations lead to the same final answer in over $80\%$ of cases. For those above $0.9$, the agreement rate reaches $100\%$.

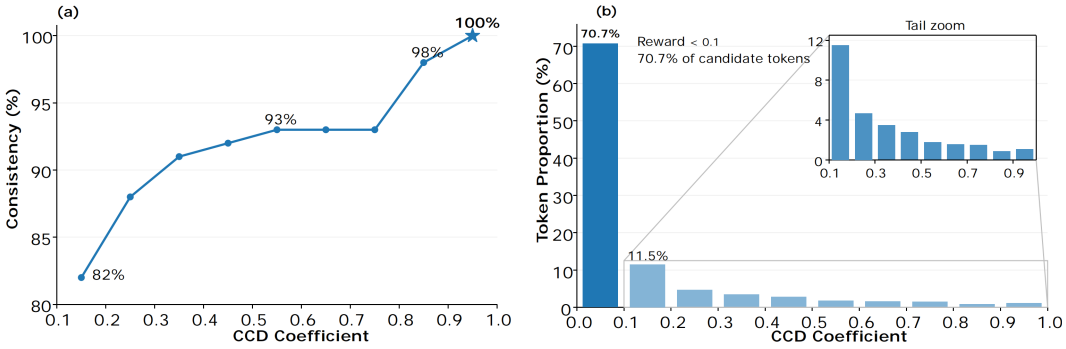



Figure 3: Left: Agreement rate of final answers between generation conditioned on reference token versus candidate token, grouped by CCD coefficient range. Right: The distribution of CCD coefficients among the top-10 tokens. Most tokens have a CCD coefficient below 0.1. This indicates that only a very small fraction of tokens at any position can lead to a valid answer.

Table 3: Time per training step (s). CCD adds only a small overhead across all settings.

| Setting | Baseline | +CCD | Overhead |
|---|---|---|---|
| Qwen3-1.7-Base, GRPO | 146.81 | 149.18 | +1.6% |
| Qwen3-4B-Base, GRPO | 271.24 | 278.05 | +2.5% |
| Qwen3-8B-Base, GRPO | 107.87 | 112.04 | +3.9% |
| Qwen3-4B-Base, SFT | 41.67 | 42.86 | +2.9% |
| Qwen3-8B-Base, SFT | 68.12 | 68.77 | +1.0% |

Figure 3 visualizes the relationship between the CCD coefficient and final answer agreement. The high consistency provides direct empirical evidence that CCD's reward is not a heuristic but a faithful proxy for contextual interchangeability. **For a more intuitive illustration, we provide concrete examples in Appendix E.**

### 5.4 Computational Overhead

CCD introduces additional computation only at the vocabulary level. The number of tokens $|\mathcal{V}|$ is negligible compared to the model's parameter count, which is on the order of billions. Moreover, we restrict the reward computation to the top-$k$ tokens, so the extra FLOPs per step scale as $k \cdot d$ rather than $|\mathcal{V}| \cdot d$, which is a tiny fraction of the forward and backward passes through a billion-parameter model. In terms of memory, the top-$k$ restriction means we only store $k$ extra scalars per token position, rather than a full vocabulary-sized tensor. For a sequence length of 8192 and $k = 10$, the extra storage per sequence is approximately $8192 \times 10 \times 2 \approx 160$ KB in bf16 precision, which is negligible compared to the multi-GB memory footprint of a billion-parameter model. Table 3 reports the time per training step.

## 6 Conclusion

We introduced **Contextual Credit Diffusion (CCD)**, a general training paradigm that diffuses the credit of a single reference token to other tokens in the vocabulary based on their contextual predictive proximity to the reference. This offers a potential solution to the one-hot supervision problem in SFT and the sparse credit problem in RL without requiring an additional teacher model. A substitution experiment shows that tokens with high CCD coefficients lead to the same final answers as the reference in most cases, confirming that CCD captures genuine contextual interchangeability and rewards multiple valid reasoning paths. Across six mathematical reasoning benchmarks, CCD consistently outperforms strong baselines in both SFT and RL with negligible overhead. Since CCD operates at the level of next-token prediction, it can be plugged into SFT, GRPO, PPO, or any method that optimizes token-level decisions, opening a new direction beyond the single-reference paradigm toward contextual predictive exploration.

### AI USE STATEMENT

In this work, we used generative AI tools for the following required-disclosure tasks: implement methods, clean and reformat dataset, interpret results, support qualitative and thematic data analysis, generate synthetic data sets and assist with translation.

We have not used generative AI tools for the following other required-disclosure tasks that were relevant to our research but performed entirely by the authors: help develop theoretical models or conceptual frameworks, propose or refine hypotheses, design or provide feedback on research methodology or experiments.

The remaining required-disclosure tasks are not applicable to this work because our study did not involve them: assist in the writing of proofs, provide critical ingredients for proving mathematical claims and formulate mathematical claims.

Additionally, we used generative AI tools for the following recommended-disclosure tasks: draft parts of a research paper, edit a research paper to improve readability, format references, and create or edit software code (for utility scripts and visualisation functions).

We have reviewed all AI-assisted work. Specifically:

- All AI-generated code was tested on multiple runs, verified for correctness against expected outputs, and peer-reviewed by at least two authors.
- The cleaned dataset was manually inspected for consistency and missing-value handling after AI-assisted preprocessing.
- AI-drafted text was substantially revised to match our scientific voice and checked for factual accuracy against our experimental results.

We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI. The core research ideas, experimental design, and final interpretation of results were performed solely by the authors without AI involvement.

### ETHICS STATEMENT

This paper studies post-training algorithms for LLMs. It does not involve human subjects, personal data, or privacy-sensitive information. All datasets used are publicly available and contain mathematical problems. We foresee no ethical concerns and declare no conflicts of interest.

### REPRODUCIBILITY STATEMENT

We describe our method in detail in Section 4. Full implementation details, including hyperparameters and training configurations are provided in Appendix D.

## REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Yijie Chen, Yijin Liu, and Fandong Meng. Sed-sft: Selectively encouraging diversity in supervised fine-tuning. *ArXiv*, abs/2602.07464, 2026. URL https://api.semanticscholar.org/CorpusID:285452829.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *ArXiv*, abs/2501.17161, 2025. URL https://api.semanticscholar.org/CorpusID:275932560.

Muxi Diao, Lele Yang, Wuxuan Gong, Yutong Zhang, Zhonghao Yan, Yufei Han, Kongming Liang, Weiran Xu, and Zhanyu Ma. Entropy-adaptive fine-tuning: Resolving confident conflicts to mitigate forgetting. *arXiv preprint arXiv:2601.02151*, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Hasan Abed Al Kader Hammoud, Kumail Alhamoud, Abed Hammoud, Elie Bou-Zeid, Marzyeh Ghassemi, and Bernard Ghanem. Train long, think short: Curriculum learning for efficient reasoning. *arXiv preprint arXiv:2508.08940*, 2025.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*, 2025.

Yuelin Hu, Zhengxue Cheng, Wei Liu, and Li Song. Entropy-gated selective policy optimization:token-level gradient allocation for hybrid training of large language models. *ArXiv*, abs/2602.03309, 2026. URL https://api.semanticscholar.org/CorpusID:285275621.

Chengyu Huang, Zhengxin Zhang, and Claire Cardie. Hapo: Training language models to reason concisely via history-aware policy optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 31122–31130, 2026.

Jerry M. Huang, Peng Lu, and Qiuhao Zeng. Calibrated language models and how to find them with label smoothing. *ArXiv*, abs/2508.00264, 2025. URL https://api.semanticscholar.org/CorpusID:280416780.

Yuxian Jiang, Yafu Li, Guanxu Chen, Dongrui Liu, Yu Cheng, and Jing Shao. Rethinking entropy regularization in large reasoning models. *arXiv preprint arXiv:2509.25133*, 2025.

Lingjing Kong, Xin Liu, Guan-Hong Chen, Martin Q. Ma, Xiangchen Song, Yuekai Sun, Mikhail Yurochkin, Taylor W. Killian, Ruslan Salakhutdinov, Kun Zhang, Eric P. Xing, and Zhengzhong Liu. From reasoning traces to reusable modules: Understanding compositional generalization in language model reasoning. 2026. URL https://api.semanticscholar.org/CorpusID:289626139.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026.

Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Ruoyu Sun, and Zhi-Quan Luo. Entropic distribution matching for supervised fine-tuning of LLMs: Less overfitting and better diversity. In *NeurIPS 2024 Workshop on Fine-Tuning in Modern Machine Learning: Principles and Scalability*, 2024a. URL https://openreview.net/forum?id=dulz3WVhMR.

Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Ruoyu Sun, and Zhimin Luo. Preserving diversity in supervised fine-tuning of large language models. In *International Conference on Learning Representations*, 2024b. URL https://api.semanticscholar.org/CorpusID:272146685.

Wenze Lin, Zhen Yang, Xitai Jiang, Pony Ma, and Gao Huang. Thickening-to-thinning: Reward shaping via human-inspired learning dynamics for llm reasoning. *arXiv preprint arXiv:2602.04265*, 2026.

Runze Liu, Jiashun Liu, Xu Wan, Yuqian Fu, and Ling Pan. When rl fails after sft: Rejuvenating model plasticity for robust sft-to-rl handoff. 2026a. URL https://api.semanticscholar.org/CorpusID:289131897.

Tao Liu, Taiqiang Wu, Runming Yang, Shaoning Sun, Junjie Wang, and Yujiu Yang. Profit: Leveraging high-value signals in sft via probability-guided token selection. *ArXiv*, abs/2601.09195, 2026b. URL https://api.semanticscholar.org/CorpusID:284717970.

Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. Learn to reason efficiently with adaptive length-based reward shaping. *arXiv preprint arXiv:2505.15612*, 2025.

Outongyi Lv, Yanzhao Zheng, Yuanwei Zhang, Zhenghao Huang, Xingjun Wang, Baohua Dong, Hangcheng Zhu, and Yingda Chen. Which tokens matter? adaptive token selection for rlvr with the relative surprisal index. 2026. URL https://api.semanticscholar.org/CorpusID:289689918.

Clara Meister, Elizabeth Salesky, and Ryan Cotterell. Generalized entropy regularization or: There's nothing special about label smoothing. In *Annual Meeting of the Association for Computational Linguistics*, 2020. URL https://api.semanticscholar.org/CorpusID:218487494.

Rui Ming, Haoyuan Wu, Shoubo Hu, Zhuolun He, and Bei Yu. One-token rollout: Guiding supervised fine-tuning of llms with policy gradient. *arXiv preprint arXiv:2509.26313*, 2025.

Jinlong Pang, Na Di, Zhaowei Zhu, Jiaheng Wei, Hao Cheng, Chen Qian, and Yang Liu. Token cleaning: Fine-grained data selection for llm supervised fine-tuning. *ArXiv*, abs/2502.01968, 2025. URL https://api.semanticscholar.org/CorpusID:276107563.

Aleksei Petrenko, Ben Lipkin, Kevin Chen, Erik Wijmans, Marco Cusumano-Towner, Raja Giryes, and Philipp Krähenbühl. Entropy-preserving reinforcement learning. *arXiv preprint arXiv:2603.11682*, 2026.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. *arXiv preprint arXiv:2604.00626*, 2026.

Zhenpeng Su, Leiyu Pan, Minxuan Lv, Yuntao Li, Wenping Hu, Fuzheng Zhang, Kun Gai, and Guorui Zhou. Ce-gppo: Coordinating entropy via gradient-preserving clipping policy optimization in reinforcement learning. *ArXiv*, abs/2509.20712, 2025. URL https://api.semanticscholar.org/CorpusID:281526201.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, et al. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong hui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *ArXiv*, abs/2506.01939, 2025. URL https://api.semanticscholar.org/CorpusID:279119146.

Yongliang Wu, Yizhou Zhou, Zhou Ziheng, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu, Lu Qi, Ming-Hsuan Yang, and Xu Yang. On the generalization of sft: A reinforcement learning perspective with reward rectification. *arXiv preprint arXiv:2508.05629*, 2025.

Violet Xiang, Chase Blagden, Rafael Rafailov, Nathan Lile, Sang Truong, Chelsea Finn, and Nick Haber. Just enough thinking: Efficient reasoning with adaptive length penalties reinforcement learning. *arXiv preprint arXiv:2506.05256*, 2025.

Kai Yang, Xin Xu, Yangkun Chen, Weijie Liu, Jiafei Lyu, Zichuan Lin, Deheng Ye, and Saiyong Yang. Entropic: Towards stable long-term training of llms via entropy stabilization with proportional-integral control. *arXiv preprint arXiv:2511.15248*, 2025.

Matthew YR Yang, Hao Bai, Ian Wu, Gene Yang, Amrith Setlur, and Aviral Kumar. Int: Self-proposed interventions enable credit assignment in llm reasoning. *arXiv preprint arXiv:2601.14209*, 2026a.

Yuchen Yang, Wenze Lin, Enhao Huang, Zhixuan Chu, Hong Zhou, Lana Tao, Yiming Li, Zhan Qin, and Kui Ren. Explainable token-level noise filtering for llm fine-tuning datasets. *ArXiv*, abs/2602.14536, 2026b. URL https://api.semanticscholar.org/CorpusID:285615674.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Honglin Yu, Weinan Dai, Yuxuan Song, Xiang Wei, Haodong Zhou, Jingjing Liu, Wei Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yong-Xu Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale. *ArXiv*, abs/2503.14476, 2025. URL https://api.semanticscholar.org/CorpusID:277104124.

# A Related Work

**Reinforcement Learning for LLMs** Reinforcement learning has become central to LLM post-training, with PPO and GRPO as dominant methods for reasoning tasks where verifiable outcomes provide natural rewards. Recent improvements focus on length normalization (Liu et al., 2025; Sui et al., 2025; Xiang et al., 2025; Hammoud et al., 2025; Hou et al., 2025; Huang et al., 2026; Lin et al., 2026), entropy control (Petrenko et al., 2026; Yang et al., 2025; Jiang et al., 2025; Su et al., 2025), and token selection (Wang et al., 2025; Lv et al., 2026; Hu et al., 2026) to improve performance. Almost all existing RL methods share a common blind spot: they apply the reward signal exclusively to the sampled token, ignoring the rich semantic structure encoded in the rest of the vocabulary. CCD addresses this by assigning rewards to every token based on its contextual interchangeability to the sampled action, making it a plug-in enhancement compatible with any policy optimization method without requiring additional models.

**Supervised Fine-Tuning** Supervised fine-tuning maximizes the log-likelihood of ground-truth tokens, but recent works find this objective encourages memorization rather than reasoning (Chu et al., 2025; Liu et al., 2026a; Kong et al., 2026). Existing methods try to improve SFT through token selection (Pang et al., 2025; Yang et al., 2026b; Liu et al., 2026b), entropy regularization (Li et al., 2024a; Chen et al., 2026; Li et al., 2024b; Diao et al., 2026), label smoothing (Meister et al., 2020; Huang et al., 2025), and RL-inspired approaches such as DFT (Wu et al., 2025) and OTR (Ming et al., 2025), which reformulate SFT as a token-level decision problem. Yet all these methods treat every non-reference token identically. This uniform treatment discards the continuous semantic structure of the vocabulary. CCD instead assigns a dense reward that reflects each token's contextual interchangeability to the reference, preserving the nuanced relationships that reasoning tasks depend on.

# B Ablation Study and Training Dynamics

## B.1 Ablation Study of $\tau$

We conduct the $\tau$ ablation on Qwen3-1.7B-Base and Qwen3-4B-Base with top-$k = 10$ fixed. Table 4 and Figures 4 and 5 report the results. CCD consistently outperforms GRPO across all values of $\tau$. On Qwen3-1.7B-Base, CCD achieves its best average accuracy of 25.0 at $\tau = 5$, compared to 20.6 for GRPO; on Qwen3-4B-Base, the best average accuracy is 33.8 at $\tau = 5$, compared to 30.9 for GRPO.

We attribute the effect of $\tau$ primarily to its impact on entropy. The effect depends on the sign of the advantage. For positive trajectories, a smaller $\tau$ makes the proximity coefficients of non-reference tokens smaller, so the positive advantage is concentrated more heavily on the reference token. This sharpens the distribution and lowers entropy. For negative trajectories, the opposite holds: a smaller $\tau$ leaves non-reference tokens only weakly suppressed, while the reference token receives the full negative advantage. Through softmax normalization, the relative probabilities of the non-reference

Table 4: Ablation study of $\tau$ on Qwen3-1.7B-Base and Qwen3-4B-Base with top-$k = 10$ (avg@16, %).

| Model | Method | AIME24 | AIME25 | AMC | MATH500 | Minerva | OlympiadBench | Avg |
|---|---|---|---|---|---|---|---|---|
| Qwen3-1.7B-Base | GRPO | 5.0 | 2.6 | 25.7 | 58.4 | 11.3 | 20.8 | 20.6 |
| | GRPO+CCD ($\tau = 1$) | 10.3 | 3.3 | 26.7 | 64.2 | 14.3 | 22.5 | 23.6 |
| | GRPO+CCD ($\tau = 5$) | 14.0 | 4.2 | 28.9 | 64.7 | 14.6 | 23.5 | 25.0 |
| | GRPO+CCD ($\tau = 10$) | 10.0 | 3.3 | 24.4 | 63.2 | 13.6 | 23.4 | 23.0 |
| Qwen3-4B-Base | GRPO | 10.6 | 13.1 | 40.3 | 74.2 | 17.2 | 30.0 | 30.9 |
| | GRPO+CCD ($\tau = 1$) | 11.9 | 14.0 | 44.9 | 75.1 | 20.6 | 34.5 | 33.5 |
| | GRPO+CCD ($\tau = 5$) | 11.4 | 15.2 | 45.3 | 75.2 | 21.1 | 34.4 | 33.8 |
| | GRPO+CCD ($\tau = 10$) | 11.0 | 11.0 | 46.8 | 75.4 | 20.0 | 35.1 | 33.2 |

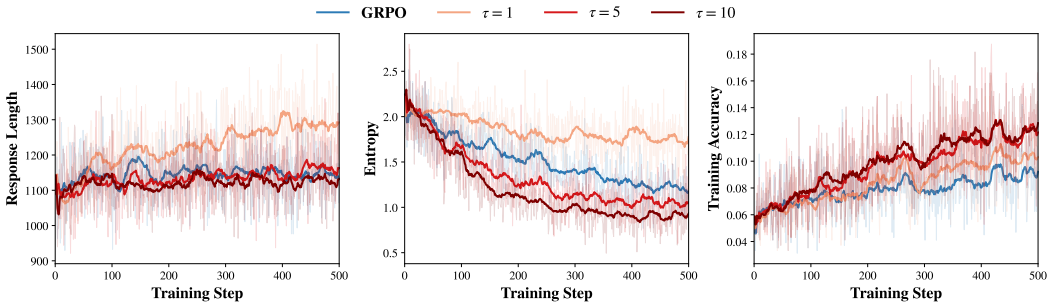



Figure 4: Ablation study of $\tau$ on Qwen3-1.7B-Base with top-$k = 10$. From left to right: response length, entropy, and training accuracy.

tokens can then rise, keeping entropy high. Since neither Qwen3-1.7B-Base nor Qwen3-4B-Base achieves high training accuracy, their training is dominated by negative trajectories, which explains why entropy increases as $\tau$ decreases in our experiments.

We further derive a practical recipe for tuning $\tau$: for a strong model whose entropy grows uncontrollably, $\tau$ should be lowered to concentrate the reward on the reference token; for a weak model whose entropy grows uncontrollably, $\tau$ should be raised to avoid the negative-trajectory effect that inflates entropy.

### B.2 Ablation Study of Top-$k$

Table 5 reports the top-$k$ ablation on Qwen3-1.7B-Base and Qwen3-4B-Base with $\tau = 5.0$ fixed. CCD with $k = 10$ and $k = 20$ performs similarly and both outperform GRPO, while $k = 100$ is slightly worse. Figure 6 and Figure 7 show the response length, entropy, and training accuracy. The three CCD variants behave similarly in the early stage, with response length stable across all $k$ and consistently lower than GRPO. Entropy decreases steadily for $k = 10$ and $k = 20$, while $k = 100$ keeps a higher and more volatile entropy throughout training. For training accuracy, $k = 10$ and $k = 20$ improve steadily, whereas $k = 100$ begins to drop in the later stage. This suggests that only a small fraction of tokens at each position can actually substitute for the reference and lead to the same correct answer, and $k = 10$ is already sufficient to cover them. Increasing $k$ to 100 does not help and even harms performance. We thus use $k = 10$ in main experiments.

## C Comparing with On-Policy Distillation

Within the next-token decision framework, OPD assigns a token-level reward equal to the log-ratio of teacher and student probabilities:

$$r_{\text{OPD}}(a, s_t) = \log \frac{\pi_{\text{teacher}}(a \mid s_t)}{\pi_\theta(a \mid s_t)}.$$

This requires an additional teacher model to provide the reference distribution. In contrast, CCD assigns a reward based on the contextual interchangeability to the reference token.

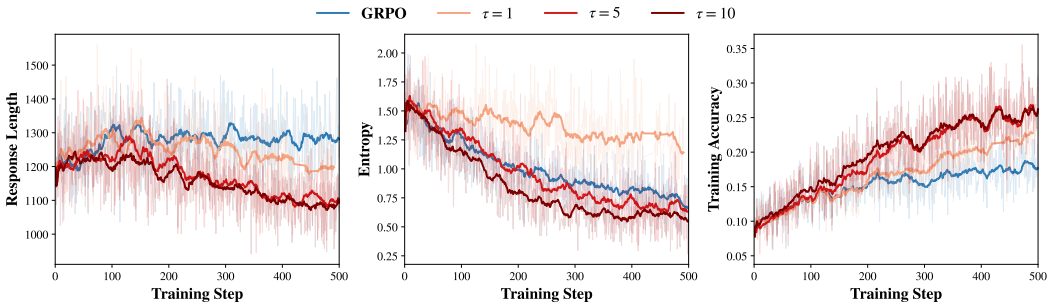



Figure 5: Ablation study of $\tau$ on Qwen3-4B-Base with top-$k = 10$. From left to right: response length, entropy, and training accuracy.

Table 5: Ablation study of top-$k$ with $\tau = 5.0$ (avg@16, %).

| Model | Method | AIME24 | AIME25 | AMC | MATH500 | Minerva | OlympiadBench | Avg |
|---|---|---|---|---|---|---|---|---|
| Qwen3-1.7B-Base | GRPO | 5.0 | 2.6 | 25.7 | 58.4 | 11.3 | 20.8 | 20.6 |
| | CCD ($k = 10$) | 14.0 | 4.2 | 28.9 | 64.7 | 14.6 | 23.5 | 25.0 |
| | CCD ($k = 20$) | 15.2 | 5.2 | 30.7 | 63.2 | 14.6 | 22.9 | 25.3 |
| | CCD ($k = 100$) | 11.3 | 4.8 | 29.7 | 63.0 | 11.6 | 22.4 | 23.8 |
| Qwen3-4B-Base | GRPO | 10.6 | 13.1 | 40.3 | 74.2 | 17.2 | 30.0 | 30.9 |
| | CCD ($k = 10$) | 11.4 | 15.2 | 45.3 | 75.2 | 21.1 | 34.4 | 33.8 |
| | CCD ($k = 20$) | 10.2 | 12.7 | 49.0 | 76.1 | 20.0 | 35.2 | 33.9 |
| | CCD ($k = 100$) | 10.0 | 16.3 | 44.9 | 74.4 | 20.1 | 34.0 | 33.3 |

We empirically compare CCD against On-Policy Distillation (OPD). Following the setup in prior work, we use Qwen3-4B-Base as the teacher model and Qwen3-1.7B-Base as the student model (Li et al., 2026). The teacher is first fine-tuned with GRPO on the DAPO-MATH-17K dataset for 3 epochs (660 steps). CCD trains the student model directly with CCD under the same GRPO framework, without any teacher involvement. Table 6 reports the results.

We stress that this comparison is not intended to show that CCD is superior to OPD. The effectiveness of OPD can vary substantially with the choice of teacher, including the teacher's capacity, quality, and degree of compatibility with the student. Our purpose here is instead to examine whether CCD, which requires no external teacher, can serve as a practical and competitive alternative.

# D Implementation Details

## D.1 Hyperparameters

We provide the detailed hyperparameter configurations used in our experiments in Table 7. All models are trained using the **Verl** (Sheng et al., 2025) framework with the settings specified below.

Table 7: Hyperparameter settings.

| Hyperparameter | Value |
|---|---|
| Learning Rate | 1e-6 |
| Train Batch Size | 256 |
| PPO Mini-Batch Size | 256 |
| Max Response Length | 8192 |
| Max Prompt Length | 1024 |
| Rollout Temperature | 1.0 |
| Evaluation Temperature | 0.7 |
| Evaluation Top-$p$ | 0.95 |

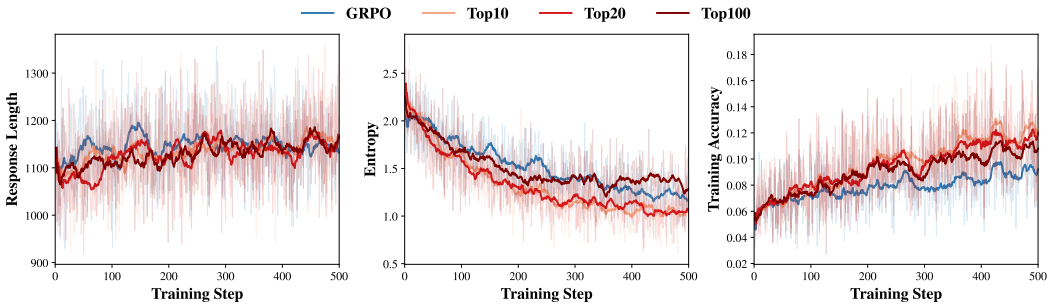



Figure 6: Ablation study of top-$k$ on Qwen3-1.7B-Base with $\tau = 5.0$. From left to right: response length, entropy, and training accuracy

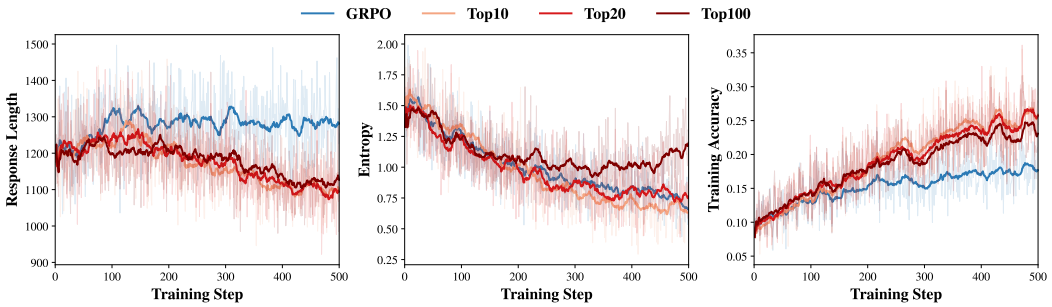



Figure 7: Ablation study of top-$k$ on Qwen3-4B-Base with $\tau = 5.0$. From left to right: response length, entropy, and training accuracy

### D.2 Baseline Details

For SFT experiments, we compare against the following baselines:

- **SFT**: Standard supervised fine-tuning with cross-entropy loss.
- **DFT** (Dynamic Fine-Tuning) (Wu et al., 2025): A token-level RL objective that assigns reward 1 to the reference token and 0 to all others. DFT introduces no additional hyperparameters; it multiplies the cross-entropy loss at each token by the model's own prediction probability.
- **EAFT** (Entropy-Adaptive Fine-Tuning) (Diao et al., 2026): A method that reweights SFT objectives with token probabilities. We set the top-$K$ entropy to $K = 20$ with a normalization factor of $\ln(20) \approx 3.0$, following the original paper.
- **Label Smoothing** (Szegedy et al., 2016): A regularization technique that smooths the one-hot label distribution. We set the smoothing factor to $\epsilon = 0.1$.

## E Case Study

We present two qualitative examples to illustrate how CCD distributes rewards across the vocabulary in practice. Figure 8 shows an RL example, and Figure 9 shows an SFT example.

**RL Example.** In the RL case (Figure 8), the reference token is "Therefore" at a step where the model is about to draw a conclusion. CCD assigns the highest CCD coefficients to "So" (0.472), "Thus" (0.400), and "Hence" (0.125), all of which are logical connectives that can substitute for "Therefore" without changing the meaning of the reasoning step. In contrast, tokens such as "However", "This", and "Equ" receive very low CCD coefficients, because they cannot serve the same logical role in this context. This demonstrates that the CCD coefficient captures the semantic interchangeability of tokens under the current context.

Table 6: Comparison with On-Policy Distillation (OPD) on Qwen3-1.7B using Qwen3-4B as teacher.

| Method | AIME24 | AIME25 | AMC | MATH500 | Minerva | Olympiad |
|---|---|---|---|---|---|---|
| Qwen3-1.7B-Base (Student) | 4.1 | 1.7 | 23.2 | 48.9 | 8.9 | 17.1 |
| Qwen3-4B-Base-GRPO (Teacher) | 10.6 | 13.1 | 40.3 | 74.2 | 17.2 | 30.0 |
| OPD | 7.7 | 3.3 | 26.4 | 61.9 | 12.0 | 22.6 |
| GRPO+CCD | 14.0 | 4.2 | 28.9 | 64.7 | 14.6 | 23.5 |

**SFT Example.** In the SFT case (Figure 9), the reference token is "specific" in the phrase "a specific given pair". CCD assigns a high CCD coefficient to "pair" (0.607), suggesting that the model considers "pair" to be a compatible continuation with the reference token in this local context. Other near-synonyms such as "given", "particular", and "arbitrary" also receive non-zero CCD coefficients, while unrelated tokens such as "student", "individual", and "random" receive near-zero CCD coefficients. This example shows that CCD assigns higher coefficients to tokens that are semantically compatible with the reference, even when they are not exactly the same word.

Together, these examples confirm that tokens with high CCD coefficients are genuinely substitutable for the reference token in the given context. They also align with our quantitative substitution experiment, which shows that replacing the reference with a high-coefficient candidate preserves the final answer in most cases.

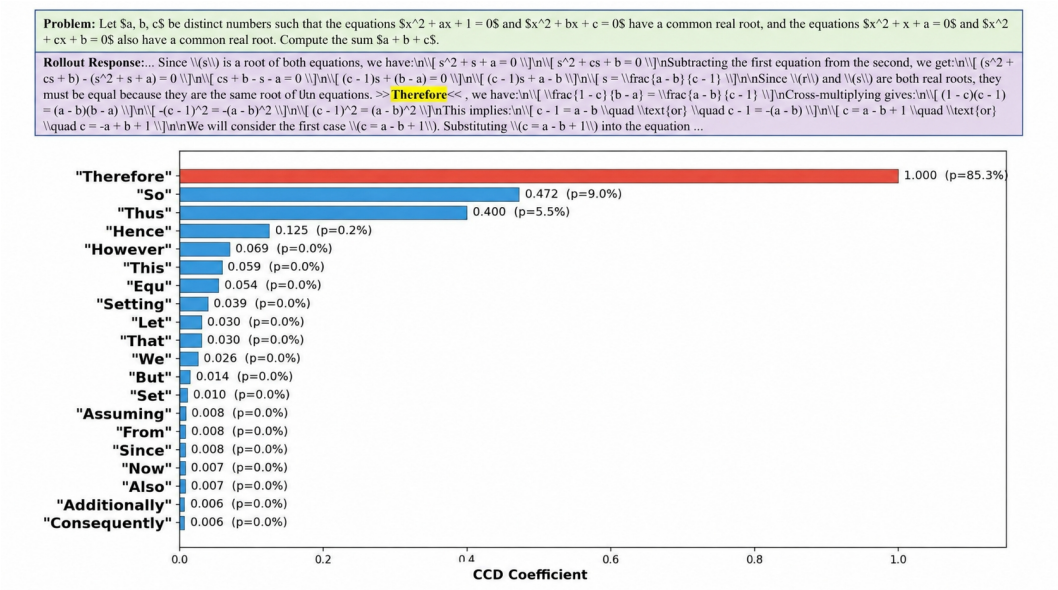



Figure 8: CCD coefficient distribution at a single token position in an RL example. The reference token is the action sampled from the model's rollout, marked with '»' and '«' (highlighted in yellow).

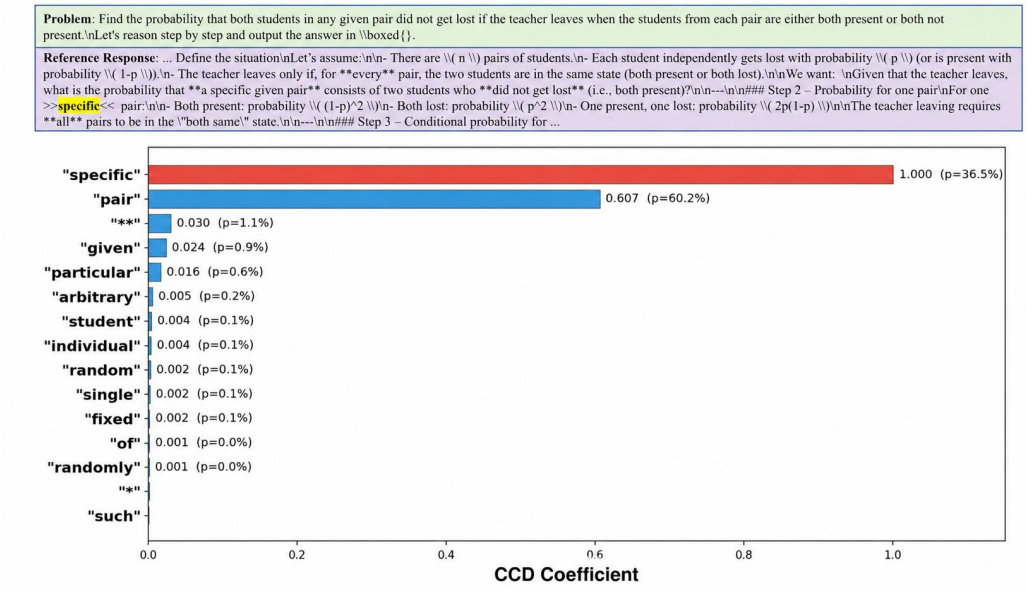



Figure 9: CCD coefficient at a single token position in an SFT example. The reference token is the ground-truth token from the dataset, marked with '»' and '«' (highlighted in yellow)