MS-TTRL: BREAKING THE FREQUENCY CONFLICT IN MULTIPLE-SOLUTION TEST-TIME REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Scientific, engineering, and mathematical reasoning tasks can admit multiple valid solutions, while exhaustive annotation of the full solution set is often prohibitively costly. Test-time reinforcement learning (TTRL) applies reinforcement learning to unlabeled test instances, offering a label-free approach to multiple-solution reasoning. In this setting, generation frequency serves two opposing objectives: correctness estimation favors frequent outcomes, whereas outcome-level diversity preservation favors less frequent outcomes. We term this tension the **Frequency Conflict**. To resolve this conflict, we propose MS-TTRL, which obtains frequency-independent correctness estimates through Outcome-Level Self-Verification while reserving generation frequency solely for diversity preservation. Three-State Outcome Cache and the Candidate State support verification reuse across repeated observations and confidence-aware handling of ambiguous outcomes, while Confidence-Gated Inverse Probability Scaling selectively strengthens the learning signals of rare, reliably verified outcomes. Experiments across two benchmarks, four multiple-solution tasks, and three backbone models, each evaluated with five random seeds, show that MS-TTRL improves Recovery Rate by 19.13 percentage points on average over the best-performing label-free baseline and approaches the performance of the supervised IPS-GRPO.

1 INTRODUCTION

Reasoning over complex scientific, engineering, and mathematical problems is central to scientific discovery, mechanistic understanding, and the development of new solutions. Many of these problems are inherently underdetermined, with limited evidence or constraints admitting multiple valid explanations, structures, or solutions (Chen et al., 2025a). Large language models are increasingly being explored as flexible solvers for such problems (Li et al., 2025; Guo et al., 2025). Reinforcement learning from verifiable rewards (RLVR), in particular, provides a direct way to adapt their behavior using verifiable task feedback (Shao et al., 2024; Hu et al., 2026a). However, standard RLVR typically relies on external supervision, such as human annotations or oracle-provided correctness labels. In multiple-solution settings, the space of valid solutions can be large and heterogeneous, making exhaustive enumeration and annotation costly or even infeasible in practice. By deriving training signals from the model's own generations, test-time reinforcement learning (TTRL) extends RLVR to unlabeled test instances (Zuo et al., 2026), making it an appealing paradigm for multiple-solution problems where complete external supervision is unavailable.

However, existing TTRL methods have been developed primarily for single-solution reasoning, where training rewards are typically estimated from the frequency of model-generated answers (Zuo et al., 2026; Wang et al., 2026a; Du et al., 2026; Yan et al., 2026; Mehrish & Vascon, 2026). Specifically, multiple responses are sampled for each test instance, and the most frequent answer is treated as a pseudo-label to reward matching reasoning trajectories (Zuo et al., 2026). In single-solution tasks, this strategy aligns well with the task structure: different valid reasoning paths typically converge to the same correct answer (Wang et al., 2022), allowing generation frequency to serve as a useful proxy for outcome correctness. Recent methods such as TTRL-CoCoV further promote diversity across reasoning trajectories while retaining frequency-based correctness estimation (Li

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.

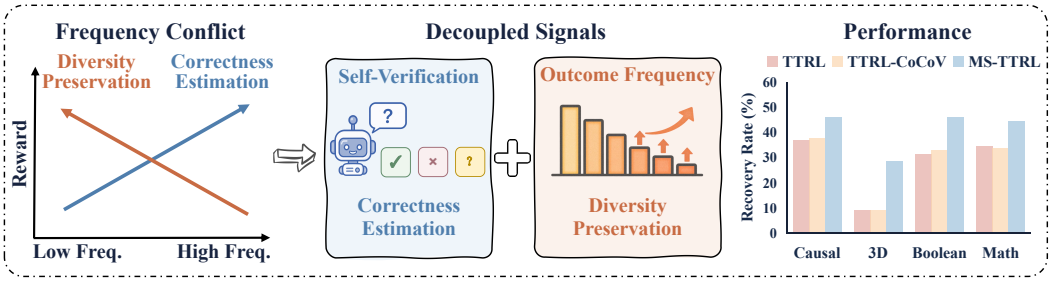


Figure 1: **Frequency conflict and signal decoupling in MS-TTRL. Left:** Frequency-based correctness estimation favors common outcomes, whereas diversity preservation favors rare ones. **Middle:** Self-verification estimates correctness, while outcome frequency promotes diversity. **Right:** MS-TTRL achieves higher Recovery Rate than TTRL and TTRL-CoCoV across four tasks.

et al., 2026a). This remains compatible with single-solution reasoning because diverse trajectories can still converge to the same final outcome.

This compatibility breaks down in multiple-solution reasoning, where diversity must also be preserved across distinct valid outcomes (Chen et al., 2025a; Sinha et al., 2026; Puri et al., 2026). In this setting, generation frequency is forced to serve two opposing objectives: correctness estimation favors frequent outcomes, whereas outcome-level diversity preservation favors underrepresented ones (Sinha et al., 2026; Li et al., 2026b; Anschel et al., 2025). Consequently, a low-frequency outcome receives conflicting interpretations: it may be regarded as unreliable by the correctness estimator, yet simultaneously prioritized for preserving outcome diversity. This ambiguity is fundamental because rare valid outcomes and rare erroneous outcomes may exhibit similarly low generation frequencies, making frequency alone insufficient to distinguish between them. We refer to this tension between correctness estimation and outcome diversity preservation as the **Frequency Conflict**.

The key to resolving the Frequency Conflict is to decouple correctness estimation from diversity preservation. We achieve this by using outcome-level self-verification as an independent correctness signal (Weng et al., 2023; Hong et al., 2024; Ma et al., 2025), while reserving generation frequency exclusively for measuring outcome rarity and diversity reweighting. Each distinct outcome is treated as a hypothesis to be verified: the model reasons backward from the candidate outcome, derives the conditions implied by it, and checks their consistency with the evidence, rules, and constraints of the problem. The resulting verification pass rate, computed as the fraction of independent verification trials that accept an outcome, serves as a confidence estimate of its validity. This design allows rare but reliably verified outcomes to receive stronger diversity-aware learning signals without similarly amplifying rare erroneous ones. Importantly, the novelty lies not in self-verification itself, but in using it to disentangle the two competing roles of generation frequency in multiple-solution TTRL. To the best of our knowledge, we are the first to systematically study the **Frequency Conflict** in label-free multiple-solution TTRL and to address it by explicitly decoupling correctness estimation from outcome diversity preservation.

Building on this decoupling principle, we propose Multiple-Solution Test-Time Reinforcement Learning (MS-TTRL), which integrates three components into a unified learning framework. MS-TTRL first obtains frequency-independent correctness estimates through *Outcome-Level Self-Verification*. To make repeated verification efficient, *Three-State Outcome Cache* stores and reuses verification results for previously observed outcomes. Verification uncertainty is handled by the *Candidate State*, which retains ambiguous outcomes, assigns soft rewards according to verification confidence, and re-verifies them when the corresponding problem is revisited. Finally, *Confidence-Gated Inverse Probability Scaling* strengthens the learning signals of rare but reliably verified outcomes while preventing uncertain or erroneous ones from being amplified.

Our main contributions are summarized as follows:

[Problem Formulation]. We identify the Frequency Conflict in multiple-solution TTRL, where generation frequency induces opposing preferences for correctness estimation and outcome diversity preservation, with frequent outcomes favored by correctness estimation and rare outcomes favored by diversity preservation. To the best of our knowledge, we are the first to characterize this conflict and explicitly decouple these two objectives in this setting.

[Algorithmic Innovation]. We propose MS-TTRL, a Multiple-Solution TTRL framework that resolves the Frequency Conflict by using outcome-level self-verification for correctness estimation while reserving generation frequency exclusively for diversity preservation. To make this decoupling efficient and reliable in practice, MS-TTRL further supports verification reuse, confidence-aware treatment of ambiguous outcomes, and gated diversity reweighting.

[Experimental Performance]. We evaluate MS-TTRL on four multiple-solution tasks from two benchmarks, using three backbone models and five random seeds. Across all evaluated settings, MS-TTRL improves Recovery Rate by an average of **19.13** percentage points over the best label-free TTRL baseline, while achieving performance close to the supervised IPS-GRPO reference.

2 Related Work

Our work connects three lines of research. **Test-time reinforcement learning (TTRL).** TTRL optimizes policies on unlabeled test instances using self-generated supervision, typically constructing rewards from agreement among sampled responses. Existing methods primarily address single-solution reasoning, improving pseudo-label reliability and balancing exploration with exploitation (Zuo et al., 2026; Liu et al., 2025; Li et al., 2026a). **Trajectory-level diversity.** In single-solution reasoning, diversity-preserving methods use Pass@K optimization and exploration-enhancing objectives to promote alternative reasoning trajectories leading to the same final answer (Chen et al., 2025b; Peng et al., 2025; Hu et al., 2026b). Broader trajectory exploration, however, does not by itself ensure coverage of distinct valid outcomes. **Outcome-level diversity.** In multiple-solution reasoning, outcome-level diversity methods promote distinct valid outcomes through inverse-probability scaling, distribution matching, or multi-answer objectives (Sinha et al., 2026; Li et al., 2026b; Puri et al., 2026). These methods rely on reliable correctness signals to guide diversity optimization. We study multiple-solution TTRL without ground-truth labels or a reliable external verifier, decoupling correctness estimation from frequency-based diversity preservation to address the Frequency Conflict. Detailed related work appears in Appendix D.

3 Frequency Conflict Between Correctness and Diversity

We consider test-time reinforcement learning for multiple-solution problems. For each problem $x \in D$, let $\mathcal{Z}^*(x)$ denote its finite set of valid outcomes. Given G sampled responses $\{y_i\}_{i=1}^G$, an outcome mapping ϕ induces the observed outcome set $\mathcal{Z}(x) = \{\phi(y_i)\}_{i=1}^G$ and empirical frequencies

$$n(z \mid x) = \sum_{i=1}^{G} \mathbf{1}[\phi(y_i) = z], \qquad z \in \mathcal{Z}(x). \tag{1}$$

Two roles of generation frequency. Frequency-based TTRL uses the most frequent outcome as a correctness pseudo-label, whereas inverse-frequency weighting favors underrepresented outcomes to preserve diversity. The corresponding scores are

$$\begin{aligned} \text{Correctness estimation:} \quad & r_{\text{corr}}(z \mid x) = \mathbf{1}[z = \hat{z}(x)], \\ \text{Diversity preservation:} \quad & r_{\text{div}}(z \mid x) = \frac{1}{n(z \mid x)}, \end{aligned} \tag{2}$$

where $\hat{z}(x) \in \arg\max_{z \in \mathcal{Z}(x)} n(z)$, with ties broken by a fixed rule. Here, r_{corr} scores pseudo-label agreement, while r_{div} scores outcome rarity. Inverse-frequency weighting promotes coverage when applied to valid outcomes. Without validity labels, however, rarity alone cannot distinguish a valid alternative from an error.

Definition 1 (Frequency Conflict)**.** For a problem x, frequency-based correctness estimation and diversity preservation exhibit a *Frequency Conflict* if there exist $z_a, z_b \in \mathcal{Z}(x)$ such that

$$\begin{gathered} n(z_a \mid x) > n(z_b \mid x), \\ \underbrace{r_{\text{corr}}(z_a \mid x) \geq r_{\text{corr}}(z_b \mid x)}_{\text{correctness favors } z_a}, \qquad \underbrace{r_{\text{div}}(z_a \mid x) \leq r_{\text{div}}(z_b \mid x)}_{\text{diversity favors } z_b}. \end{gathered} \tag{3}$$

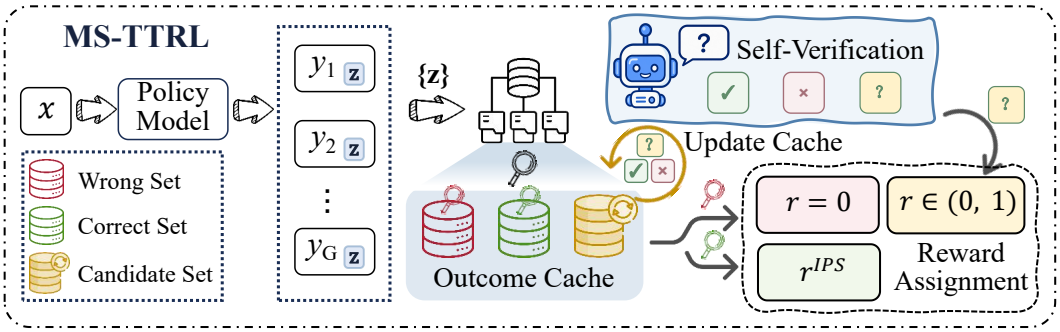


Figure 2: **Overview of MS-TTRL.** For each unlabeled problem, generated solution outcomes are self-verified and maintained in a three-state outcome cache. Their stored status is then combined with gated inverse-probability scaling to reward outcomes, thereby decoupling correctness estimation from diversity preservation with limited cost of self-verification.

For the scores in Eq. 2, this conflict arises whenever an observed outcome is less frequent than the selected mode: it receives a zero correctness score but a higher diversity score. Frequency alone cannot establish whether this outcome is a valid but rare alternative that should be promoted or an error that should be suppressed. This ambiguity motivates estimating correctness independently of generation frequency. Section 5.4 shows that extending pseudo-label selection to the top-k outcomes can not alleviate this conflict.

4 Method

MS-TTRL resolves the Frequency Conflict by separating correctness estimation from frequency-based diversity optimization. It consists of three components: outcome-level self-verification, three-state outcome cache with an uncertainty buffer, and confidence-gated inverse probability scaling. Figure 2 illustrates how these components interact, while Algorithm 1 in Appendix A provides the complete training procedure.

4.1 Outcome-Level Self-Verification

To resolve the Frequency Conflict, MS-TTRL estimates correctness through self-verification and reserves generation frequency for diversity preservation. Rare valid outcomes can thus receive positive correctness signals and diversity promotion.

The same model serves as actor and verifier, evaluating final outcomes rather than complete responses or intermediate reasoning. This shortens the context and avoids repeated verification of equivalent responses within a group.

To improve verification reliability, a reverse-reasoning prompt treats each outcome as a hypothesis: the verifier derives its implied conditions under the task rules and checks them against the problem. At step t, we map the G responses sampled for x to outcomes through $z = \phi(y)$ and merge duplicates to obtain

$$\mathcal{Z}_t(x) = \{z_1, z_2, \ldots, z_M\}, \qquad M \leq G. \tag{4}$$

For each outcome z, we perform n independent verification trials and define its verification pass rate as

$$p_t(z \mid x) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}[\text{Verify}_i(z, x) = 1], \tag{5}$$

where $\text{Verify}_i(z, x) = 1$ denotes acceptance by trial i. The pass rate $p_t(z \mid x)$ serves as the correctness estimate throughout optimization.

4.2 THREE-STATE OUTCOME CACHE

Repeatedly verifying resolved outcomes is unnecessary and expensive. For each problem x, MS-TTRL therefore maintains a persistent cache indexed by the problem–outcome pair (x, z). The cache partitions all previously observed outcomes into a correct set $\mathcal{S}_t(x)$, a candidate set $\mathcal{C}_t(x)$, and a wrong set $\mathcal{W}_t(x)$. These sets represent verifier-assigned states rather than ground-truth labels.

When problem x is revisited, only unresolved candidates and newly observed outcomes require verification. The outcomes verified at step t are

$$\mathcal{U}_t(x) = \mathcal{C}_{t-1}(x) \cup \left[\mathcal{Z}_t(x) \setminus \left(\mathcal{S}_{t-1}(x) \cup \mathcal{C}_{t-1}(x) \cup \mathcal{W}_{t-1}(x)\right)\right]. \tag{6}$$

Previously resolved correct and wrong outcomes retain their cached states without additional verifier calls.

For each $z \in \mathcal{U}_t(x)$, its new state is determined by

$$\mathrm{State}_t(z \mid x) = \begin{cases} \texttt{correct}, & p_t(z \mid x) \geq \tau_{\text{high}}, \\ \texttt{candidate}, & \tau_{\text{low}} < p_t(z \mid x) < \tau_{\text{high}}, \\ \texttt{wrong}, & p_t(z \mid x) \leq \tau_{\text{low}}, \end{cases} \tag{7}$$

where $0 \leq \tau_{\text{low}} < \tau_{\text{high}} \leq 1$. The corresponding cache entry stores the assigned state and the latest verification score. Correct and wrong outcomes are resolved and reused in later steps, whereas candidates remain unresolved and are re-verified when the problem is revisited.

Candidate state as an uncertainty buffer. A binary cache would have to accept or reject an outcome after a single verification round. Premature acceptance introduces erroneous positive rewards, whereas premature rejection may suppress rare valid outcomes before the verifier becomes sufficiently reliable. The candidate state postpones this decision. Meanwhile, candidates retain policy support through their latest verification scores $p_t(z \mid x)$: they are neither treated as confirmed correct outcomes nor assigned the zero reward used for wrong outcomes. The candidate state therefore acts as both a verification buffer and an exploration buffer.

4.3 CONFIDENCE-GATED INVERSE PROBABILITY SCALING

Following IPS-GRPO, we scale the base reward to strengthen learning signals for infrequent correct outcomes and mitigate outcome-level collapse. Let $n_t(z_i)$ be the count of outcome z_i in the current group, with empirical probability $n_t(z_i \mid x)/G$. We restrict scaling to the correct set and control its strength with λ to prevent scaled rewards from overwhelming candidate soft rewards:

$$\tilde{r}_t(z_i) = \begin{cases} \lambda + (1-\lambda)\frac{G}{n_t(z_i|x)}, & z_i \in \mathcal{S}_t(x), \\ p_t(z_i \mid x), & z_i \in \mathcal{C}_t(x), \\ 0, & z_i \in \mathcal{W}_t(x). \end{cases} \tag{8}$$

The coefficient $\lambda \in [0, 1]$ interpolates between full IPS ($\lambda = 0$) and the unscaled base reward ($\lambda = 1$).

This gating mechanism is particularly important in the label-free setting: applying inverse-probability scaling to candidate outcomes would amplify low-frequency verification noise. Candidate outcomes therefore retain policy support through soft rewards, whereas diversity amplification is restricted to outcomes assigned to the correct set.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

We evaluate MS-TTRL on three HypoSpace subsets (Chen et al., 2025a)—Causal Discovery, 3D Structure Reconstruction, and Boolean Expression Discovery—and on MACE-Math (Wang et al., 2026b), an independent multiple-solution mathematical reasoning benchmark. Each problem admits a finite but potentially varying set of valid outcomes. Training uses only unlabeled problems; complete solution sets are reserved exclusively for evaluation.

Table 1: Main results over 5 random seeds on the three HypoSpace subsets. RR and ORR denote Recovery Rate and Outcome Recovery Rate, respectively. Higher is better for both metrics. **IPS-GRPO, as shown in light gray, is a label-supervised reference rather than a baseline**. "Best" denotes the better-performing result between TTRL and TTRL-CoCoV for each metric.

Method	**Causal** (@16)		**3D** (@27)		**Boolean** (@20)	
	RR(%) ↑	**ORR(%)** ↑	**RR(%)** ↑	**ORR(%)** ↑	**RR(%)** ↑	**ORR(%)** ↑
Qwen3-1.7B						
IPS-GRPO †	51.20 ± 0.70	28.19 ± 0.53	27.75 ± 0.15	16.35 ± 0.04	46.15 ± 0.41	37.38 ± 0.57
TTRL	36.94 ± 0.00	17.42 ± 0.00	9.07 ± 0.00	5.28 ± 0.00	31.03 ± 0.13	22.63 ± 0.13
TTRL-CoCoV	37.54 ± 0.11	17.63 ± 0.04	9.07 ± 0.00	5.28 ± 0.00	32.88 ± 1.38	25.11 ± 0.74
MS-TTRL	45.89 ± 0.49	23.26 ± 0.20	28.39 ± 0.04	16.93 ± 0.03	46.00 ± 1.10	36.49 ± 0.51
Δ(MS-TTRL−Best)	$\mathbf{+8.35}$	$\mathbf{+5.63}$	$\mathbf{+19.32}$	$\mathbf{+11.65}$	$\mathbf{+13.12}$	$\mathbf{+11.38}$
Qwen3-8B						
IPS-GRPO †	65.23 ± 0.69	40.96 ± 0.82	54.51 ± 0.58	43.24 ± 0.44	46.46 ± 1.41	37.72 ± 0.5
TTRL	36.94 ± 0.00	17.42 ± 0.00	9.07 ± 0.00	5.28 ± 0.00	26.03 ± 0.80	16.43 ± 0.22
TTRL-CoCoV	36.91 ± 0.00	17.42 ± 0.00	9.48 ± 0.27	5.54 ± 0.09	26.78 ± 0.62	19.28 ± 0.45
MS-TTRL	61.65 ± 0.44	38.32 ± 0.44	57.13 ± 0.40	45.35 ± 0.63	47.95 ± 0.53	37.75 ± 0.33
Δ(MS-TTRL−Best)	$\mathbf{+24.71}$	$\mathbf{+20.90}$	$\mathbf{+47.65}$	$\mathbf{+39.81}$	$\mathbf{+21.17}$	$\mathbf{+18.47}$
Phi4-4B-mini-Instruct						
IPS-GRPO †	45.28 ± 0.44	23.44 ± 0.26	26.98 ± 0.03	15.73 ± 0.02	30.20 ± 1.30	21.08 ± 1.04
TTRL	35.90 ± 0.00	16.29 ± 0.00	0.64 ± 0.00	0.55 ± 0.00	9.66 ± 0.05	6.69 ± 0.13
TTRL-CoCoV	36.94 ± 0.00	17.42 ± 0.00	1.36 ± 0.13	1.16 ± 0.09	6.55 ± 0.73	5.69 ± 1.06
MS-TTRL	42.03 ± 0.10	23.10 ± 0.04	26.94 ± 0.10	15.55 ± 0.16	26.10 ± 1.26	19.80 ± 0.60
Δ(MS-TTRL−Best)	$\mathbf{+5.09}$	$\mathbf{+5.68}$	$\mathbf{+25.58}$	$\mathbf{+14.39}$	$\mathbf{+16.44}$	$\mathbf{+13.11}$

All methods use the same evaluation budget, set to the largest valid-solution count in each task. Recovery Rate (RR) averages coverage across problems, whereas Outcome Recovery Rate (ORR) aggregates recovered outcomes over all valid outcomes and therefore gives more weight to problems with larger solution sets.

We compare against two label-free baselines: **TTRL**, which uses majority-vote pseudo-labels, and **TTRL-CoCoV**, which additionally promotes trajectory-level diversity. **IPS-GRPO** uses ground-truth correctness labels and serves only as a supervised reference.

HypoSpace experiments use Qwen3-1.7B, Qwen3-8B, and Phi4-4B-mini-Instruct; MACE-Math uses Qwen3-1.7B. We report means and standard deviations over five random seeds. Dataset splits and implementation details are provided in Appendices B and C.

5.2 Main Results

Table 2: Performance on the MACE-Math benchmark. Higher is better for all metrics.

Method	**RR (%)** ↑	**ORR (%)** ↑
IPS-GRPO †	68.19 ± 0.49	55.26 ± 0.57
TTRL	34.39 ± 0.25	22.75 ± 0.21
TTRL-CoCoV	33.75 ± 0.25	22.34 ± 0.12
MS-TTRL	44.26 ± 0.56	34.74 ± 0.56
Δ(MS-TTRL−Best)	$\mathbf{+9.87}$	$\mathbf{+11.99}$

MS-TTRL outperforms both label-free baselines in RR and ORR across all HypoSpace tasks and backbones (Table 1), with gains that remain consistent over five random seeds. On MACE-Math, it improves RR and ORR over the stronger label-free baseline by 9.87 and 11.99 percentage points, respectively (Table 2), while remaining below the base model and supervised IPS-GRPO on both metrics. Across the three HypoSpace tasks, MS-TTRL requires approximately 23.2% more total training time than TTRL, with nearly unchanged peak GPU memory (Tables 3a and 3b). It is faster than TTRL-CoCoV on Causal and 3D and matches its reported Boolean runtime. High accuracy does not ensure broad outcome coverage: TTRL and TTRL-CoCoV approach perfect accuracy in several 3D settings while collapsing to a dominant outcome. TTRL-CoCoV does not consistently improve RR or ORR over TTRL, indicating that trajectory-level diversity alone does not prevent outcome-level collapse.

Accuracy results and further analysis appear in Appendices E.1 and F.2.

Table 3: Training time and peak GPU memory on Qwen3-1.7B across three tasks. Lower is better.

(a) Runtime

Method	Causal	3D	Boolean
IPS-GRPO	2h 25m	39m	32m
TTRL	2h 21m	38m	28m
TTRL-CoCoV	3h 03m	53m	38m
MS-TTRL	2h 51m	46m	38m

(b) Peak GPU Memory (GB)

Method	Causal	3D	Boolean
IPS-GRPO	126.94	121.44	103.66
TTRL	129.14	117.57	102.87
TTRL-CoCoV	130.40	118.98	104.60
MS-TTRL	131.83	123.29	107.54

5.3 Ablation and Sensitivity Analysis

5.3.1 Ablation Study

We ablate three supporting components on Qwen3-1.7B: the cache across all three HypoSpace tasks, and the Candidate Set and Gated IPS on Causal. Appendix F.3 provides detailed interpretations.

Three-State Outcome Cache. Reusing resolved outcome states reduces self-verification samples by **75.9%** overall across the three tasks and by **96.1%** in the final epoch (Figure 3).

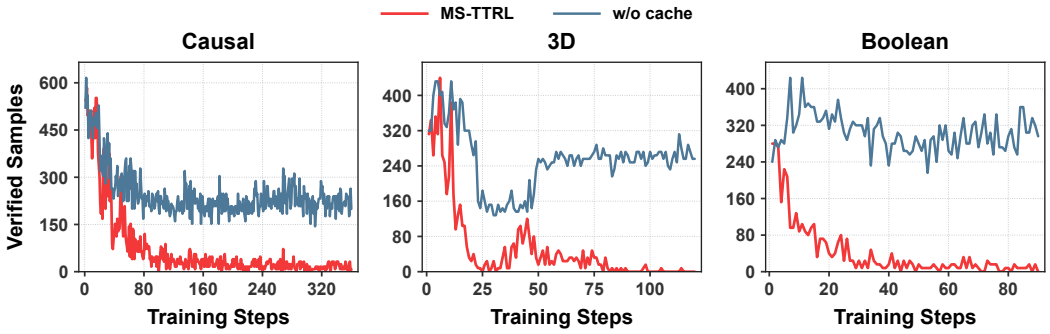


Figure 3: Ablation of the Three-State Outcome Cache on Qwen3-1.7B across the Causal, 3D, and Boolean tasks. Removing the cache mechanism substantially increases the number of verified samples throughout training, showing that reusing previously stored verification states effectively reduces repeated self-verification and lowers the additional inference cost.

Candidate-Based Uncertainty Buffer. Without the Candidate Set, outcomes receive binary labels using τ_{low}, τ_{high}, or their midpoint. On Causal, higher thresholds improve accuracy but reduce coverage, while lower thresholds produce the opposite trade-off (Table 4). Retaining uncertain outcomes allows MS-TTRL to achieve 93.75% accuracy and 45.89% RR, compared with 74.80% and 41.79% under the midpoint threshold.

Confidence-Gated Inverse Probability Scaling. Replacing Gated IPS with ungated inverse-frequency scaling reduces Causal accuracy from 93.75% to 59.59% (Table 4). The gate protects correctness by restricting amplification to reliably verified outcomes; ungated scaling can amplify low-frequency errors that receive positive rewards.

5.3.2 Hyperparameter Sensitivity Analysis

We analyze sensitivity to key hyperparameters; Appendix F.4 provides detailed interpretations.

Number of self-verification trials. Increasing verification trials from 4 to 8 improves Causal recovery; 16 trials bring a smaller additional RR gain, although ORR improves further. On 3D, four trials already give recovery close to that with eight (Table 5). We use 8 trials across tasks to balance verification reliability and cost.

Table 4: Ablation study of Gated IPS and the Candidate Set on Qwen3-1.7B for the Causal task. Results are averaged over five random seeds.

Configuration	Causal		
	ACC(%) ↑	**RR(%)** ↑	**ORR(%)** ↑
w/o Gated IPS	59.59 ± 0.68	49.46 ± 0.79	27.68 ± 0.54
w/o Candidate Set (high)	99.93 ± 0.06	37.15 ± 0.28	17.50 ± 0.10
w/o Candidate Set (mid)	74.80 ± 0.90	41.79 ± 0.77	20.55 ± 0.49
w/o Candidate Set (low)	63.59 ± 0.62	45.12 ± 0.44	23.81 ± 0.49
MS-TTRL	93.75 ± 0.46	45.89 ± 0.49	23.26 ± 0.20

Table 5: Sensitivity to the number of self-verification trials on Qwen3-1.7B for Causal and 3D Structure Reconstruction.

Verification	Causal		3D	
	RR (%) ↑	**ORR (%)** ↑	**RR (%)** ↑	**ORR (%)** ↑
4	41.54	20.64	27.08	15.63
8	45.89	23.26	28.41	16.88
16	46.40	26.13	27.11	15.78

Table 6: Sensitivity to the thresholds for outcome-state assignment on Qwen3-1.7B for the 3D Structure Reconstruction task. One threshold is varied while the other is fixed.

τ_{high}	τ_{low}	**RR (%)** ↑	**ORR (%)** ↑
Varying τ_{low} *(*$\tau_{\text{high}} = 0.875$*)*			
0.875	0.375	27.18	15.83
0.875	0.500	28.41	16.88
0.875	0.625	21.32	12.29
Varying τ_{high} *(*$\tau_{\text{low}} = 0.500$*)*			
0.750	0.500	27.16	15.83
0.875	0.500	28.41	16.88
1.000	0.500	26.14	15.67

Outcome-state thresholds. Around the default threshold setting $(\tau_{\text{high}}, \tau_{\text{low}}) = (0.875, 0.5)$, recovery on the 3D task is less sensitive to changes in τ_{high} than to increasing τ_{low} to 0.625 (Table 6). The latter leads to a substantial reduction in coverage, which is consistent with prematurely rejecting uncertain but potentially valid outcomes during outcome-state assignment.

Strength of Gated IPS. Moderate scaling ($\lambda = 0.9$) achieves the best recovery performance on the 3D task among all tested values of the scaling parameter. Compared with this moderate setting, disabling IPS ($\lambda = 1$) reduces Recovery Rate from 28.39% to 10.35%, while applying stronger scaling also results in lower recovery (Table 7).

Table 7: Sensitivity to the diversity scaling coefficient λ on Qwen3-1.7B for the 3D Structure Reconstruction task.

λ	1.00	0.95	0.90	0.85	0.80	0.75
RR (%) ↑	10.35	26.05	28.39	27.20	20.69	18.14
ORR (%) ↑	5.52	15.52	16.93	15.82	12.28	10.56

5.4 Empirical Analysis of the Frequency Conflict

We test whether treating the top-k most frequent outcomes as positive pseudo-labels resolves the Frequency Conflict. Table 8 shows that increasing k sharply degrades accuracy on Causal and 3D without reliably improving coverage, while Boolean exhibits a trade-off between lower accuracy and higher coverage.

Expanding the pseudo-label set admits both rare valid and rare invalid outcomes, so frequency-based pseudo-labels cannot reliably separate correctness from diversity. In contrast, the Self-Verify variant shows that simply decoupling correctness estimation from diversity preservation through *Outcome-level Self-Verification* improves coverage over TTRL while maintaining high accuracy. These results support decoupling correctness estimation from frequency-based diversity weighting; detailed analysis appears in Appendix F.5.

Table 8: Effect of extending TTRL from majority voting to the top-k most frequent outcomes on Qwen3-1.7B across three multiple-solution tasks. TTRL-k treats the top-k generated outcomes as positive pseudo-labels. All methods use inverse-probability scaling when assigning rewards.

	Causal (@16)			**3D** (@27)			**Boolean** (@20)		
Method	**Acc(%)** ↑	**RR(%)** ↑	**ORR(%)** ↑	**Acc(%)**	**RR(%)**	**ORR(%)**	**Acc(%)**	**RR(%)**	**ORR(%)**
Self-Verify	93.41	46.92	25.00	97.57	27.10	15.78	87.86	46.28	37.39
TTRL	99.97	36.94	17.42	99.87	9.07	5.28	81.79	31.03	22.63
TTRL-2	18.95	19.26	10.06	22.10	2.81	2.19	66.25	35.44	26.33
TTRL-3	1.29	1.73	0.94	0.00	0.00	0.00	52.14	36.80	30.61
TTRL-4	1.42	2.62	1.82	2.68	1.71	1.52	45.54	39.07	31.01

5.5 Learning Signals for Rare Valid Outcomes

On Qwen3-1.7B, we identify the least-frequent valid outcome(s) in each rollout group containing at least two distinct valid outcomes, retaining ties. Mean Advantage measures their average optimization signal, and Positive Advantage Rate measures the fraction receiving a positive signal.

Across all three HypoSpace tasks, MS-TTRL maintains positive Mean Advantage and gives most rare valid outcomes positive advantages, whereas TTRL and TTRL-CoCoV tend to suppress them (Figure 4). Supervised IPS-GRPO also promotes these outcomes. This diagnostic is consistent with the proposed mechanism for preserving rare valid solutions; detailed definitions and discussion are provided in Appendix F.6.

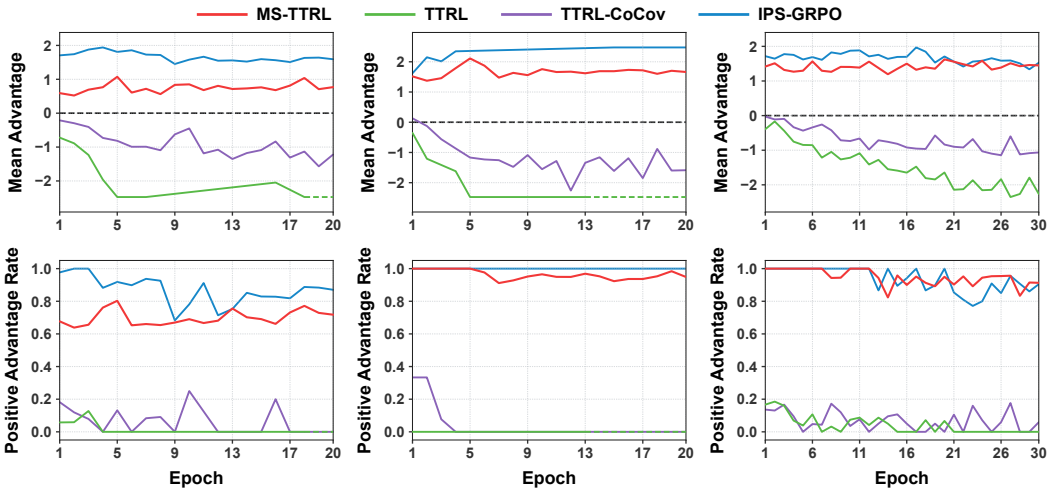


Figure 4: Learning signals assigned to rare valid outcomes on Qwen3-1.7B. For each rollout group containing at least two distinct valid outcomes, we identify the valid outcome(s) with the lowest generation frequency. Each subfigure reports the mean advantage and positive advantage rate across training epochs for one task.

6 Conclusion

We presented MS-TTRL, a label-free test-time reinforcement learning framework for multiple-solution problems. We identified the Frequency Conflict caused by using generation frequency both to estimate correctness and to preserve outcome-level diversity. MS-TTRL separates these roles through outcome-level self-verification, Three-State Outcome Cache and Confidence-Gated Inverse-Probability Scaling. Across four tasks, three backbone models, and five random seeds, MS-TTRL consistently improves recovery over label-free baselines while approaching the supervised IPS-GRPO reference. These results show that decoupling correctness estimation from diversity preservation can improve valid-outcome coverage without requiring complete solution annotations during training.

AI USE STATEMENT

We used generative AI tools to polish and revise the language of the manuscript and to assist with code development. The core research ideas were developed by the authors without generative AI assistance. We manually reviewed the AI-assisted manuscript revisions and AI-generated code, and take full responsibility for the final content of this work, including its text and code.

ETHICS STATEMENT

This work studies test-time reinforcement learning on existing reasoning benchmarks. We do not identify any ethical concerns specific to the methods or experiments presented in this paper.

REPRODUCIBILITY STATEMENT

We provide our implementation in the supplementary materials and at the following anonymous code repository: `https://anonymous.4open.science/r/Multiple-Solution-TTRL-3BFA`. Appendices B and C document the datasets, splits, training and evaluation configurations, and Appendix G provides the prompts used for Causal Discovery self-verification. The main experiments report means and standard deviations over five random seeds. Together, these materials support reproduction of the reported results.

REFERENCES

Oron Anschel, Alon Shoshan, Adam Botach, Shunit Haviv Hakimi, Asaf Gendler, Emanuel Ben Baruch, Nadav Bhonker, Igor Kviatkovsky, Manoj Aggarwal, and Gerard Medioni. Group-aware reinforcement learning for output diversity in large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 32382–32403, 2025.

Tingting Chen, Beibei Lin, Zifeng Yuan, Qiran Zou, Hongyu He, Anirudh Goyal, Yew-Soon Ong, and Dianbo Liu. Hypospace: A diagnostic benchmark for set-valued hypothesis generation under underdetermination and sublinear coverage bounds. *arXiv preprint arXiv:2510.15614*, 2025a.

Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. Pass@ k training for adaptively balancing exploration and exploitation of large reasoning models. *arXiv preprint arXiv:2508.10751*, 2025b.

Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *Advances in Neural Information Processing Systems*, 38:57654–57689, 2026.

Bodong Du, Xuanqi Huang, and Xiaomeng Li. Distribution-aware reward estimation for test-time reinforcement learning. *arXiv preprint arXiv:2601.21804*, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Anthony GX-Chen, Jatin Prakash, Jeff Guo, Rob Fergus, and Rajesh Ranganath. Kl-regularized reinforcement learning for generative modelling is designed to mode collapse. In *International Conference on Learning Representations*, volume 2026, pp. 50665–50693, 2026.

Xu He, Jialiang Guo, Fucheng Xiong, Haodong Zhao, Ke Zeng, Xunliang Cai, et al. Vane: Guiding high-value exploration in rlvr via outcome-process novelty shaping. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 28721–28739, 2026.

Ruixin Hong, Hongming Zhang, Xinyu Pang, Dong Yu, and Changshui Zhang. A closer look at the self-verification abilities of large language models in logical reasoning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 900–925, 2024.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *Advances in Neural Information Processing Systems*, 38:162239–162262, 2026a.

Zican Hu, Shilin Zhang, Yafu Li, Jianhao Elliott Yan, Xuyang Hu, Leyang Cui, Xiaoye Qu, Chunlin Chen, Yu Cheng, and Zhi Wang. Diversity-incentivized exploration for versatile reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 71175–71205, 2026b.

Jiahui Li, Jianfeng Shan, Wenpei Chen, Shunyu Wu, Jian Lou, Wenjie Feng, Dan Li, and See-Kiong Ng. Exploiting verification-generation gap: Test-time reinforcement learning with confidence-conditioned verification. *arXiv preprint arXiv:2606.03608*, 2026a.

Xiaozhe Li, Xinyu Fang, Shengyuan Ding, Linyang Li, Haodong Duan, Qingwen Liu, and Kai Chen. Np-engine: Empowering optimization reasoning in large language models with verifiable synthetic np problems. *arXiv preprint arXiv:2510.16476*, 2025.

Xiaozhe Li, Yang Li, Xinyu Fang, Shengyuan Ding, Peiji Li, Yongkang Chen, Yichuan Ma, Tianyi Lyu, Linyang Li, Dahua Lin, et al. Beyond mode collapse: Distribution matching for diverse reasoning. *arXiv preprint arXiv:2605.19461*, 2026b.

Ruotong Liao, Nikolai Röhrich, Xiaohan Wang, Yuhui Zhang, Yasaman Samadzadeh, Volker Tresp, and Serena Yeung-Levy. Tool verification for test-time reinforcement learning. *arXiv preprint arXiv:2603.02203*, 2026.

Hongxiang Lin, Zhirui Kuai, Erpeng Xue, and Lei Wang. Detecting and mitigating the correct-answer extinction window in test-time reinforcement learning with majority voting. *arXiv preprint arXiv:2605.19444*, 2026.

Jia Liu, ChangYi He, YingQiao Lin, MingMin Yang, FeiYang Shen, and ShaoGuo Liu. Ettrl: Balancing exploration and exploitation in llm test-time reinforcement learning via entropy mechanism. *arXiv preprint arXiv:2508.11356*, 2025.

Ruotian Ma, Peisong Wang, Cheng Liu, Xingyan Liu, Jiaqi Chen, Bang Zhang, Xin Zhou, Nan Du, and Jia Li. S2r: Teaching llms to self-verify and self-correct via reinforcement learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22632–22654, 2025.

Ambuj Mehrish and Sebastiano Vascon. Core: Consensus rewards via equilibrium for test-time reinforcement learning. *arXiv preprint arXiv:2608.09324*, 2026.

Teng Pan, Yuchen Yan, Zixuan Wang, Ruiqing Zhang, Guiyang Hou, Wenqi Zhang, Weiming Lu, Jun Xiao, and Yongliang Shen. Coverrl: Breaking the consensus trap in label-free reasoning via generator-verifier co-evolution. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29833–29853, 2026.

Ruotian Peng, Yi Ren, Zhouliang Yu, Weiyang Liu, and Yandong Wen. Simko: Simple pass@ k policy optimization. 2025.

Isha Puri, Mehul Damani, Idan Shenfeld, Marzyeh Ghassemi, Jacob Andreas, and Yoon Kim. Reaching beyond the mode: Rl for distributional reasoning in language models. *arXiv preprint arXiv:2603.24844*, 2026.

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.

Abhijeet Sinha, Sundari Elango, and Dianbo Liu. Expected return causes outcome-level mode collapse in reinforcement learning and how to fix it with inverse probability scaling. *arXiv preprint arXiv:2601.21669*, 2026.

Christian Walder and Deep Tejas Karkhanis. Pass@ k policy optimization: Solving harder reinforcement learning problems. *Advances in Neural Information Processing Systems*, 38:152416–152445, 2026.

Weiqin Wang, Yile Wang, Kehao Chen, and Hui Huang. Beyond majority voting: Towards fine-grained and more reliable reward signal for test-time reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 37251–37265, 2026a.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Yuhan Wang, Shiyu Ni, Zhikai Ding, Zihang Zhan, Yuanzi Li, and Keping Bi. Evaluating and calibrating llm confidence on questions with multiple correct answers. *arXiv preprint arXiv:2602.07842*, 2026b.

Yumeng Wang, Zhiyuan Fan, Jiayu Liu, Jen-tse Huang, and Yi R Fung. Diversity-enhanced reasoning for subjective questions. In *International Conference on Learning Representations*, volume 2026, pp. 75302–75332, 2026c.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. Large language models are better reasoners with self-verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 2550–2575, 2023.

Jianghao Wu, Yasmeen George, Jin Ye, Yicheng Wu, Daniel F Schmidt, and Jianfei Cai. Spine: Token-selective test-time reinforcement learning with entropy-band regularization. *arXiv preprint arXiv:2511.17938*, 2025.

Kunbin Xu, Xingzuo Li, Xuefeng Bai, and Kehai Chen. Hi-ttrl: Regulating consensus with hints for test-time reinforcement learning. *arXiv preprint arXiv:2608.03545*, 2026.

Dong Yan, Jian Liang, Yanbo Wang, Shuo Lu, Ran He, and Tieniu Tan. What if consensus lies? selective-complementary reinforcement learning at test time. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28957–28970, 2026.

Suqin Yuan, Jinkun Chen, Jiyang Zheng, Muyang Li, Lei Feng, Dadong Wang, Tao Xiang, Tongliang Liu, and Bo An. Understanding diversity collapse in rlvr via the lens of overtraining. *arXiv preprint arXiv:2606.15455*, 2026.

Fengji Zhang, Xinyao Niu, Chengyang Ying, Guancheng Lin, Zhongkai Hao, Zhou Fan, Chengen Huang, Jacky Keung, Bei Chen, and Junyang Lin. A ^{2} search: Ambiguity-aware question answering with reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 136394–136448, 2026.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Xinwei Long, Ermo Hua, Biqing Qi, et al. Ttrl: Test-time reinforcement learning. *Advances in Neural Information Processing Systems*, 38:131459–131483, 2026.

A MS-TTRL ALGORITHM

Algorithm 1 summarizes MS-TTRL. Outcome caches persist across visits to each problem, and only newly observed outcomes and cached candidates are self-verified.

Algorithm 1 Multiple-Solution Test-Time Reinforcement Learning

Require: Policy π_θ; unlabeled problems D; rollout size G; verification trials n; thresholds $\tau_{\text{high}}, \tau_{\text{low}}$; IPS coefficient λ; update budget T.

Initialize $S(x) = C(x) = W(x) = \emptyset$ for every $x \in D$
for $t = 1, \ldots, T$ **do**
  Sample a batch $B \subseteq D$
  for each $x \in B$ **do**
    Sample $\{y_i\}_{i=1}^{G} \sim \pi_\theta(\cdot \mid x)$; extract $z_i = \phi(y_i)$
    $Z_t(x) \leftarrow \{z_i\}_{i=1}^{G}$; $n_t(z) \leftarrow \sum_{i=1}^{G} \mathbf{1}[z_i = z]$
    $U_t(x) \leftarrow C(x) \cup \big(Z_t(x) \setminus (S(x) \cup W(x))\big)$
    for each $z \in U_t(x)$ **do**
      $p_t(z \mid x) \leftarrow \frac{1}{n} \sum_{j=1}^{n} \mathbf{1}[\text{Verify}_{\pi_\theta, j}(z, x) = 1]$
      if $p_t(z \mid x) \geq \tau_{\text{high}}$ **then** $S(x) \leftarrow S(x) \cup \{z\}$; $C(x) \leftarrow C(x) \setminus \{z\}$
      else if $p_t(z \mid x) \leq \tau_{\text{low}}$ **then** $W(x) \leftarrow W(x) \cup \{z\}$; $C(x) \leftarrow C(x) \setminus \{z\}$
      else $C(x) \leftarrow C(x) \cup \{z\}$; store the latest $p_t(z \mid x)$
      end if
    end for
    for $i = 1, \ldots, G$ **do**
      if $z_i \in S(x)$ **then** $\tilde{r}_i \leftarrow \lambda + (1 - \lambda) G / n_t(z_i)$
      else if $z_i \in C(x)$ **then** $\tilde{r}_i \leftarrow p_t(z_i \mid x)$
      else $\tilde{r}_i \leftarrow 0$
      end if
    end for
  end for
  Compute group-normalized advantages and update θ with GRPO on B
end for
return π_θ

B DATASET DETAILS

HypoSpace: Causal Discovery. The Causal Discovery task evaluates whether a model can recover multiple directed acyclic graphs (DAGs) that are consistent with the same intervention observations. Each problem contains four nodes, $\{A, B, C, D\}$, with up to six directed edges. Observations are generated by perturbing a subset of nodes and recording the resulting binary effects on the remaining nodes according to their causal descendants. Multiple DAGs may induce the same intervention observations and are therefore treated as distinct valid outcomes. Following the original HypoSpace construction, the complete set of admissible DAGs is exhaustively enumerated, enabling exact evaluation of solution coverage. We use 300 problems for training and 92 problems for evaluation.

HypoSpace: 3D Structure Reconstruction. The 3D Structure Reconstruction task requires the model to infer three-dimensional voxel structures from a two-dimensional top-view projection. Each problem is defined on a 3×3 grid with a maximum height of three layers. Multiple height assignments can correspond to the same top view, provided that they satisfy the physical support constraint that every voxel above the first layer is supported by a voxel directly below it. All valid height assignments consistent with the observed projection are treated as distinct outcomes. The finite solution space is exhaustively enumerated for each problem. We use 100 problems for training and 29 problems for evaluation.

HypoSpace: Boolean Expression Discovery. The Boolean Expression Discovery task considers symbolic expressions over two variables, x and y, using the operators AND, OR, and NOT. Candidate expressions are generated up to depth two and canonically deduplicated according to algebraic

equivalences such as commutativity, idempotence, and associativity. Given a partial set of input–output observations, the model is asked to recover Boolean expressions consistent with the observed behavior. Since multiple expressions can satisfy the same observations, each problem naturally admits a finite set of valid outcomes. We use 50 problems for training and 28 problems for evaluation.

MACE: Mathematical Reasoning. MACE is a multiple-answer benchmark designed to evaluate model behavior when a question admits several distinct correct answers. We use its mathematical reasoning subset, in which each problem asks the model to produce a number satisfying a specified mathematical property within a given interval, such as being a prime, square, cube, Fibonacci, or triangular number. The benchmark provides the complete set of valid answers for each problem, making it suitable for evaluating outcome-level coverage. For training, we select 250 problems with exactly six valid answers. For evaluation, we construct a separate test set by sampling 100 problems from each group corresponding to different numbers of valid answers, enabling us to assess the model's generalization across varying degrees of solution multiplicity.

To evaluate generalization and enable comparison with the supervised reference, all methods are trained on the same training split and evaluated on a held-out test split. During training, ground-truth labels are available only to IPS-GRPO and are withheld from all label-free methods.

C Training and Evaluation Details

All methods share the same training configuration unless otherwise specified. For each prompt, we sample $G = 8$ responses with temperature 1.0 and top-$p = 1.0$. The maximum prompt and response lengths are both 1024 tokens. Training uses AdamW with a learning rate of 1×10^{-6}. The PPO clipping range is $[0.2, 0.28]$. We normalize advantages by the group standard deviation following GRPO. The batch size is 16 prompts. Models are trained for 20 epochs, except for Boolean, which uses 30 epochs. The default random seed is 1337.

For IPS-GRPO, rewards are inverse-frequency scaled using the empirical outcome frequency within each rollout group. TTRL uses the most frequent outcome among the $G = 8$ responses as the pseudo-label. For TTRL-CoCoV, we use Pass@k with $k = 4$, a diversity coefficient of $\lambda_{\text{div}} = 0.05$, confidence thresholds of $0.6/0.4$, and 8 verifier samples.

For MS-TTRL, we use Gated IPS with $\lambda = 0.9$. Outcome states are determined using verification thresholds $\tau_{\text{high}} = 0.875$ and $\tau_{\text{low}} = 0.5$, with 8 verification samples for each unresolved outcome. Candidate outcomes receive their verification pass rate as a soft reward.

For evaluation, we sample k responses for each problem with temperature 1.0, where k is set to the maximum number of valid solutions among all problems in the corresponding task. We report the average trajectory-level accuracy across sampled responses, together with solution coverage measured by Recovery Rate and Outcome Recovery Rate.

D Detailed Related Work

D.1 Test-Time Reinforcement Learning

Test-time reinforcement learning (TTRL) adapts language models directly on unlabeled test instances, targeting settings in which ground-truth answers or reliable external reward signals are unavailable. Standard TTRL (Zuo et al., 2026) samples multiple responses from the current policy, selects the majority outcome as a pseudo-label, and constructs rewards by comparing individual responses against this consensus. This enables policy improvement using only the model's own generations, but also makes the quality of optimization highly dependent on the reliability of the induced pseudo-supervision.

Subsequent work has primarily focused on improving the reliability and stability of this self-generated supervision. Several methods refine consensus-based pseudo-labeling through uncertainty estimation, confidence-aware filtering, selective supervision, distribution-aware aggregation, or interventions based on training dynamics (Liu et al., 2025; Wang et al., 2026a; Du et al., 2026; Yan et al., 2026; Lin et al., 2026; Wu et al., 2025; Xu et al., 2026). Another line of work introduces self-verification or jointly improves generation and verification to identify unreliable consensus signals

before they are used for optimization (Pan et al., 2026; Li et al., 2026a; Liao et al., 2026). Despite these advances, generation frequency remains central to pseudo-label construction or confidence estimation in most existing TTRL methods. Moreover, these methods are predominantly developed for single-solution reasoning, where the main challenge is obtaining reliable supervision rather than maintaining coverage over multiple distinct valid outcomes. Our work instead studies TTRL in the multiple-solution setting, where correctness estimation must coexist with outcome-level diversity preservation.

D.2 Trajectory-Level Diversity

In conventional single-solution reasoning, different correct responses may follow substantially different reasoning trajectories while ultimately converging to the same correct outcome. Diversity in this setting therefore primarily concerns the coverage of alternative reasoning paths rather than distinct final answers. Prior studies have observed that reinforcement learning from verifiable rewards can improve Pass@1 while reducing Pass@K for larger K, suggesting that optimization may concentrate probability mass on a narrower subset of successful reasoning trajectories (Chen et al., 2026; Yuan et al., 2026).

A growing body of work addresses this trajectory-level contraction by explicitly encouraging broader exploration during reinforcement learning. Representative approaches directly optimize Pass@K-style objectives, modify advantage estimation to account for multiple sampled trajectories, or introduce exploration-oriented objectives that discourage excessive concentration on a small number of high-reward reasoning paths (Chen et al., 2025b; Walder & Karkhanis, 2026; Peng et al., 2025; Hu et al., 2026b; He et al., 2026). Related ideas have also been explored in the label-free setting. In particular, TTRL-CoCoV (Li et al., 2026a) augments TTRL with confidence-conditioned verification and exploration mechanisms to promote diversity among sampled reasoning trajectories. These methods improve exploration over alternative reasoning processes, but the resulting trajectories still converge to the same terminal outcome.

D.3 Outcome-Level Diversity

A different form of collapse arises when a problem admits multiple distinct valid outcomes. In this setting, reinforcement learning may concentrate probability mass on only a few easily discovered solutions, causing other valid outcomes to disappear from the policy distribution. The resulting objective is therefore not merely to maintain diverse reasoning trajectories, but to preserve coverage over the valid outcome space. Benchmarks such as HypoSpace provide controlled multiple-solution tasks in which such coverage can be explicitly evaluated through recovery-based metrics (Chen et al., 2025a; Li et al., 2025).

Recent work has begun to address outcome-level collapse from several complementary perspectives. Frequency-aware methods such as IPS-GRPO and related approaches mitigate the reinforcement of dominant solutions by assigning stronger learning signals to underrepresented valid outcomes (Sinha et al., 2026; Anschel et al., 2025). Distribution-based methods such as DMPO align policy probabilities with reward-defined target distributions to promote broader coverage over high-reward outcomes (Li et al., 2026b; GX-Chen et al., 2026). Other approaches directly optimize diverse output generation, for example through set-level or multi-answer objectives (Puri et al., 2026; Zhang et al., 2026; Wang et al., 2026c). Although these methods differ in optimization strategy, they share an important assumption: reliable correctness signals are available to determine which generated outcomes are valid before diversity is promoted.

Our setting removes this assumption. We study outcome-level diversity preservation when neither ground-truth solutions nor a reliable external verifier is available during training. In this setting, generation frequency simultaneously serves as a proxy for outcome correctness and as a signal of underrepresentation: high-frequency outcomes are favored by correctness estimation, whereas low-frequency outcomes are favored by diversity preservation.

E Additional Experiment Results

E.1 Accuracy Report

As shown in Tables 9 and 10, we report the accuracy results corresponding to Tables 1 and 2. MS-TTRL maintains high accuracy while achieving strong valid-solution coverage.

Table 9: Accuracy over 5 evaluation seeds on the three HypoSpace subsets. Higher is better.

Method	Causal ACC(%) ↑	3D ACC(%) ↑	Boolean ACC(%) ↑
Qwen3-1.7B			
IPS-GRPO †	95.14 ± 0.63	99.82 ± 0.13	88.54 ± 0.79
TTRL	99.97 ± 0.03	99.87 ± 0.07	81.79 ± 0.81
TTRL-CoCoV	97.73 ± 0.18	99.97 ± 0.05	77.79 ± 0.91
MS-TTRL	93.75 ± 0.46	99.13 ± 0.53	90.54 ± 0.99
Qwen3-8B			
IPS-GRPO †	96.05 ± 0.17	98.62 ± 0.34	94.86 ± 0.52
TTRL	99.92 ± 0.08	99.87 ± 0.11	84.50 ± 0.79
TTRL-CoCoV	96.49 ± 0.49	96.53 ± 0.53	89.36 ± 0.91
MS-TTRL	95.80 ± 0.18	98.72 ± 0.30	97.00 ± 0.77
Phi4-4B-mini-Instruct			
IPS-GRPO †	95.68 ± 0.29	99.77 ± 0.15	62.46 ± 2.36
TTRL	99.92 ± 0.10	10.34 ± 0.00	38.00 ± 0.13
TTRL-CoCoV	99.89 ± 0.18	18.31 ± 0.48	21.96 ± 0.84
MS-TTRL	97.42 ± 0.32	77.85 ± 0.85	45.57 ± 1.52

Table 10: Accuracy on the MACE-Math benchmark using Qwen3-1.7B. Higher is better.

Method	ACC (%) ↑
IPS-GRPO †	82.83 ± 0.64
TTRL	65.99 ± 0.29
TTRL-CoCoV	62.86 ± 0.35
MS-TTRL	67.00 ± 0.35

E.2 Training Dynamics

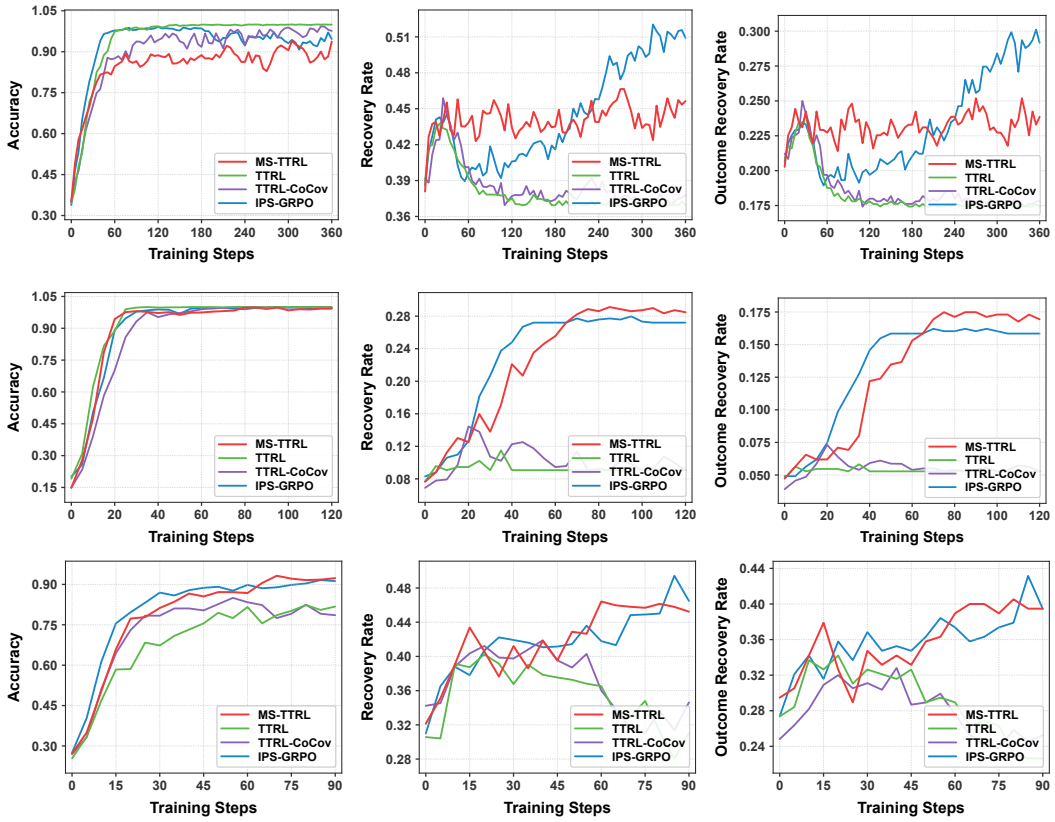
We present the training curves of Qwen3-1.7B on three subtasks of the HypoSpace Benchmark: Causal, 3D, and Boolean. As shown in figure 5, although TTRL and TTRL-CoCoV substantially improve accuracy, they also exhibit severe mode collapse. In particular, both the Recovery Rate (RR) and Outcome Recovery Rate (ORR) decline significantly during the later stages of training.

F Detailed Experimental Analysis

This appendix provides a detailed analysis supporting the experimental summaries in the main text. All associated tables and figures remain in the main text.

F.1 Evaluation Metrics

For each subset, we determine the evaluation sampling budget based on the maximum number of valid solutions admitted by a single problem, and evaluate all methods using the same budget. We report the Recovery Rate and Outcome Recovery Rate to evaluate solution-space coverage. Recovery Rate first computes the fraction of valid solutions recovered for each problem and then averages across problems, assigning equal weight to every problem. Outcome Recovery Rate instead aggregates all recovered valid solutions across the dataset and divides by the total number of valid



(a) Accuracy (b) Recovery rate (c) Outcome recovery rate

Figure 5: Training dynamics on the Causal (top), 3D (middle), and boolean (bottom) tasks. From left to right: (a) accuracy, (b) recovery rate, (c) outcome recovery rate.

solutions, assigning equal weight to every solution and consequently more weight to problems with larger solution sets. Although the two metrics typically exhibit similar trends, they emphasize average per-problem coverage and aggregate coverage of the valid solution space, respectively.

F.2 Main Results and Efficiency

As shown in Table 1, MS-TTRL consistently and substantially outperforms TTRL and TTRL-CoCoV across all three HypoSpace tasks in both Recovery Rate (RR) and Outcome Recovery Rate (ORR). On MACE-Math, MS-TTRL also achieves substantial improvements (Table 2), indicating that the method generalizes beyond HypoSpace to an independent multiple-solution reasoning benchmark.

These gains persist across model scales (Qwen3-1.7B and Qwen3-8B), model families (Qwen and Phi), and five random seeds, suggesting that they are not tied to a particular backbone or favorable sampling randomness. Accuracy results are reported in Appendix E.1. Because our primary focus is valid-outcome coverage, the main table reports RR and ORR, while accuracy is used mainly for diagnostic analyses. Self-verification introduces additional inference cost, but the Three-State Outcome Cache reuses previously verified outcome states and keeps the overhead moderate. As shown in Table 3a, MS-TTRL incurs approximately 23.2% additional wall-clock training time over TTRL, remains faster than TTRL-CoCoV on Causal and 3D, and matches its reported runtime on Boolean. Its peak GPU memory allocation remains nearly unchanged relative to both methods (Table 3b), indicating that cache reuse amortizes repeated verification.

The results further reveal that high trajectory-level correctness does not necessarily imply broad solution-space coverage. TTRL and TTRL-CoCoV can achieve near-perfect trajectory-level accu-

racy in several settings while concentrating on only a small subset of valid outcomes. This collapse is particularly pronounced on the 3D Structure Reconstruction task: TTRL and TTRL-CoCoV with Qwen3-1.7B, and TTRL with Qwen3-8B, answer the problems correctly but effectively produce a single dominant outcome.

Improving trajectory-level diversity likewise does not necessarily improve outcome-level diversity. Although TTRL-CoCoV encourages exploration of alternative reasoning trajectories, its gains over vanilla TTRL in RR and ORR remain limited across most model-task configurations. Thus, diversifying reasoning paths toward the same answer does not by itself broaden coverage of distinct valid outcomes, motivating direct optimization at the outcome level.

MS-TTRL also remains competitive with supervised IPS-GRPO and occasionally achieves higher recovery in several settings, despite using no ground-truth solutions during training. One possible explanation is that HypoSpace tasks are relatively tractable for identifying valid solutions, while the main challenge is preserving diversity among them; under these conditions, exact correctness labels may provide limited additional benefit for recovery. MS-TTRL also assigns soft positive rewards to outcomes whose correctness is uncertain, some of which may be invalid. Although this introduces noise, it can preserve a broader exploratory distribution and occasionally improve solution-space coverage.

F.3 Component Ablations

Three-State Outcome Cache. We first examine whether the Three-State Outcome Cache reduces the additional inference cost introduced by self-verification. As shown in Figure 3, reusing stored verification states substantially reduces repeated self-verification across all three tasks. The cache reduces the total number of self-verification samples by **75.9%** over the three tasks, with the reduction reaching **96.1%** in the final epoch. These results show that it amortizes self-verification by avoiding redundant checks of previously resolved outcomes.

Candidate-Based Uncertainty Buffer. We next ablate the Candidate-Based Uncertainty Buffer to study its role in preventing premature misclassification of uncertain outcomes. Without the Candidate Set, each outcome must be assigned an immediate binary correct/incorrect state using a single verification threshold. We consider three choices for this threshold: τ_{low}, τ_{high}, and their midpoint $(\tau_{\text{high}} + \tau_{\text{low}})/2$. As shown in Table 4, the midpoint causes a substantial accuracy drop on the Causal task. Raising the threshold improves accuracy but reduces RR and ORR, whereas lowering it has the opposite effect. A high threshold can falsely reject valid but insufficiently verified outcomes, causing them to lose positive rewards and reducing coverage. Conversely, a low threshold more readily accepts incorrect outcomes as valid, improving exploration at the cost of accuracy. The Candidate-Based Uncertainty Buffer avoids this forced trade-off by retaining ambiguous outcomes until additional verification evidence becomes available.

Confidence-Gated Inverse Probability Scaling. Finally, we evaluate the necessity of gating inverse-probability scaling by replacing Gated IPS with the original IPS rule, which applies inverse-frequency amplification without conditioning on the verification state. As shown in Table 4, removing the gate causes a sharp drop in Causal accuracy. In the label-free setting, a positive reward does not guarantee correctness: low-frequency incorrect outcomes can receive positive rewards and then be disproportionately amplified by vanilla IPS. This amplifies noisy rewards and can increase the policy probability of erroneous outcomes. Gated IPS prevents this failure mode by restricting amplification to outcomes verified with sufficiently high confidence, thereby promoting underrepresented valid outcomes without similarly amplifying uncertain or erroneous ones.

F.4 Hyperparameter Sensitivity

Number of self-verification trials. As shown in Table 5, the verification budget needed for a reliable correctness estimate varies across tasks. On Causal Discovery, increasing the number of trials from 4 to 8 clearly improves RR and ORR; increasing it to 16 yields a smaller additional RR gain but further improves ORR. On 3D Structure Reconstruction, four trials already achieve near-optimal recovery. Thus, additional self-verification improves outcome-level correctness estimation,

but its marginal benefit diminishes after a sufficient budget is reached. Because verification also incurs inference cost, we use 8 trials by default as a balance between reliability and efficiency.

Outcome-state thresholds. Table 6 studies the two outcome-state thresholds, τ_{high} and τ_{low}. Around the default setting of $\tau_{\text{high}} = 0.875$ and $\tau_{\text{low}} = 0.5$, varying τ_{high} has only a minor effect on RR and ORR, indicating that assigning high-confidence outcomes to the Correct Set is relatively robust. Performance is more sensitive to τ_{low}: decreasing it to 0.375 causes little degradation, whereas increasing it to 0.625 noticeably reduces both recovery metrics. An overly large lower threshold may prematurely assign valid but temporarily uncertain outcomes to the Wrong Set. Because their cached states are then reused and their rewards become zero, such false rejections can progressively suppress their generation probability and reduce coverage. This asymmetry highlights the importance of retaining an intermediate Candidate Set.

Strength of Gated IPS. Finally, Table 7 examines the Gated IPS coefficient λ. Setting $\lambda = 1.0$ disables inverse-probability scaling and substantially reduces both RR and ORR, showing that correctness rewards alone are insufficient to prevent outcome-level collapse and that explicit amplification of underrepresented valid outcomes is important for recovery. Performance also deteriorates when λ is too small, corresponding to overly aggressive scaling. Because IPS is applied only to Correct-Set outcomes, excessive scaling enlarges the reward gap between confirmed outcomes and uncertain Candidate outcomes, weakening the latter during exploration and potentially reducing diversity. Overall, λ around 0.9 provides the best balance between reliable correctness signals and promotion of underrepresented valid outcomes.

F.5 Frequency Conflict Analysis

A natural alternative to MS-TTRL is to retain frequency-based pseudo-labeling while expanding the positive set from the single majority outcome to the top-k most frequent outcomes. We evaluate whether admitting more low-frequency outcomes resolves the Frequency Conflict between correctness estimation and outcome-level diversity preservation.

As shown in Table 8, enlarging the pseudo-label set does not reliably preserve both correctness and coverage; instead, it produces two failure modes.

First, Causal and 3D exhibit optimization collapse: accuracy is high at $k = 1$ but drops rapidly as k increases and approaches collapse for larger k. The expanded set treats more low-frequency invalid outcomes as positive, while inverse-probability scaling gives those outcomes stronger learning signals. Erroneous pseudo-labels are therefore both rewarded and amplified, allowing noisy supervision to dominate optimization.

Second, Boolean exhibits a clear accuracy–coverage trade-off. Small k preserves higher accuracy but excludes more low-frequency valid outcomes; larger k improves coverage by admitting more rare outcomes but also increases the chance of rewarding low-frequency invalid outcomes and thus degrades accuracy.

These results show that the Frequency Conflict cannot be resolved by simply increasing the number of pseudo-labels. As long as frequency determines both correctness and diversity rewards, changing k only shifts the boundary between excluding rare valid outcomes and admitting rare invalid ones. Self-verification provides an independent correctness signal, while frequency can be reserved for diversity preservation. As a minimal realization, we perform n independent verification trials per outcome and use the pass rate as its correctness reward; this decoupling already improves coverage while maintaining high accuracy.

F.6 Learning Signals for Rare Valid Outcomes

We further investigate whether MS-TTRL promotes underrepresented valid outcomes in the current rollout distribution. We conduct this analysis on Qwen3-1.7B across the Causal, 3D, and Boolean tasks. For each training epoch, we consider problems whose rollout groups contain at least two distinct valid outcomes and identify, among the valid outcomes observed for each problem, the outcome with the lowest generation frequency.

We report Mean Advantage and Positive Advantage Rate. Mean Advantage is the average advantage assigned to underrepresented valid outcomes during policy optimization and indicates whether the update tends to promote or suppress them. Positive Advantage Rate is the fraction of rare valid outcomes receiving a positive advantage and measures how consistently they are promoted. We compare TTRL, TTRL-CoCoV, IPS-GRPO, and MS-TTRL; IPS-GRPO is included only as a supervised reference because it uses ground-truth validity labels.

As shown in Figure 4, MS-TTRL consistently provides positive learning signals to rare valid outcomes: its Mean Advantage remains positive, and most rare valid outcomes receive positive advantages. TTRL and TTRL-CoCoV, whose pseudo-labels are based on generation frequency, instead exhibit a clear suppressive effect. IPS-GRPO, which can directly identify valid outcomes using ground-truth labels, also consistently promotes underrepresented valid outcomes and therefore serves as a supervised reference upper bound.

These results show that self-verification identifies low-frequency valid outcomes without using generation frequency as a correctness proxy. This provides direct evidence that MS-TTRL maintains broader valid-outcome coverage than TTRL and TTRL-CoCoV under label-free training.

G Prompt Templates and Self-Verification Example

We provide the prompts used in the Causal Discovery task to make the outcome-level self-verification procedure concrete. The first prompt specifies the task and asks the policy model to generate a candidate causal graph. The second prompt asks the verifier to check one candidate graph against the observations, and the final block shows a representative verifier response.

G.1 Causal Task Input Prompt

```
You are given observations from perturbation experiments on a causal
    system.

    Semantics:
    - When a node is perturbed, the perturbed node is 0.
    - A node is 1 if it is a downstream descendant of the perturbed
    node in the causal graph.
    - All other nodes are 0.

    Nodes: A, B, C, D
Constraint: The graph should have at most 6 edges.

    Observations:
    Perturb(B) -> A:0 B:0 C:0 D:0
    Perturb(C) -> A:0 B:1 C:0 D:1
    Perturb(D) -> A:0 B:1 C:0 D:0

    Task:
    Output a single directed acyclic graph (DAG) over the nodes above
    that explains all observations.
    Before producing the final answer, reason through the observations
    step by step.
    The final answer should follow the formatting rules.

    Formatting rules:
    1) Use only the listed nodes. No self-loops. No cycles.
    2) Respond with exactly one line:
    - If there are edges: Graph: A->B, B->C
    - If there are no edges: Graph: No edges
```

G.2 SELF-VERIFICATION PROMPT

```
Verify whether the candidate causal graph explains the perturbation
    observations.

Semantics:
- When a node is perturbed, the perturbed node is 0.
- A node is 1 if it is a downstream descendant of the perturbed node
    in the causal graph.
- All other nodes are 0.

Nodes: A, B, C, D

Constraint: The graph should have at most 6 edges.

Reasoning example:
An edge `X->Y` points from the upstream cause X to the downstream
    effect Y.
Follow arrows only in this left-to-right direction; do not reverse an
    edge.
For candidate graph `A->B, B->D`, B and D are descendants of A, while
    C is not.
Therefore, perturbing A gives A:0, B:1, C:0, D:1, so
`Perturb(A) -> A:0 B:1 C:0 D:1` is consistent. In contrast,
`Perturb(A) -> A:0 B:0 C:0 D:1` is inconsistent because B is reachable
    from A.

Perturbation observations:
Perturb(B) -> A:0 B:0 C:0 D:0
Perturb(C) -> A:0 B:1 C:0 D:1
Perturb(D) -> A:0 B:1 C:0 D:0

Candidate graph:
B->A, B->C, B->D, C->D

Reason through reachability for every listed perturbation. Judge only
    this candidate; multiple candidates may be valid.

First provide a reasoning paragraph. Then end your response with
    exactly one line:
Verification Result: True
or
Verification Result: False
```

G.3 Representative Self-Verification Output

```
To verify whether the candidate graph explains the perturbation
    observations, we need to check if the observed results match the
    expected values under the given causal graph.

The candidate graph is: **B->A, B->C, B->D, C->D**.

Let's analyze each perturbation:

1. **Perturb(B)**:
   - B is perturbed, so B:0.
   - B is a direct ancestor of A (B->A), so A is a downstream
    descendant of B. Thus, A:1.
   - C is a direct descendant of B (B->C), so C:1.
   - D is a descendant of C (C->D), so D:1.
   - Result: A:1, B:0, C:1, D:1.
   - Observed result: A:0, B:0, C:0, D:0.
   - **Inconsistent**. The observed result differs from what the graph
    predicts.

2. **Perturb(C)**:
   - C is perturbed, so C:0.
   - A is not a descendant of C (C->D, but A is not reachable from C),
    so A:0.
   - B is not a descendant of C, so B:0.
   - D is a descendant of C (C->D), so D:1.
   - Result: A:0, B:0, C:0, D:1.
   - Observed result: A:0, B:1, C:0, D:1.
   - **Inconsistent**. The observed result differs from what the graph
    predicts.

3. **Perturb(D)**:
   - D is perturbed, so D:0.
   - A is not a descendant of D, so A:0.
   - B is not a descendant of D, so B:0.
   - C is not a descendant of D, so C:0.
   - Result: A:0, B:0, C:0, D:0.
   - Observed result: A:0, B:1, C:0, D:0.
   - **Inconsistent**. The observed result differs from what the graph
    predicts.

Since the perturbation of B results in inconsistent observations, the
    candidate graph does **not** explain the perturbation observations.

Verification Result: False
```