ADAPTIVE CONTRASTIVE SELF-PLAY FOR REGIME-AWARE REINFORCEMENT LEARNING FINE-TUNING OF LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) relies on variation among sampled rewards. When every response in a rollout group receives effectively the same shaped verifier reward, Group Relative Policy Optimization (GRPO) receives no policy gradient from reward for that group. We ask whether the optimization objective should change as such groups become common. We introduce Adaptive Contrastive Self-Play (A-CSP), which monitors the fraction of zero variance groups and switches between GRPO and a contrastive objective that learns from higher and lower reward responses when eligible pairs exist.

Across 156 training runs on GSM8K and a custom MATH-L1-3 pool, we find that the value of switching is strongly regime dependent. On Qwen2.5-1.5B GSM8K, A-CSP reaches similar peak exact match to GRPO, 0.730 versus 0.737, while retaining more of its peak performance at the end of training, 0.97 versus 0.90. On Qwen2.5-3B, A-CSP improves peak accuracy on both tasks and final accuracy on GSM8K, while final accuracy on MATH-L1-3 is essentially unchanged. Mechanistic controls show that masking KL on zero variance groups does not reproduce the A-CSP behavior, while the decomposition suggests that batch composition affects optimization dynamics more than KL masking alone on the flagship setting. Simple switching schedules can also match or exceed A-CSP in peak accuracy, showing that zero variance alone is not sufficient to determine the best switching policy. These results show that adaptive objective switching can improve RLVR training in some regimes, but its benefit depends on more than the prevalence of zero variance groups.

1 INTRODUCTION

Reinforcement learning has become a central tool for adapting language models to human preferences and task feedback Christiano et al. (2017); Ziegler et al. (2019); Stiennon et al. (2020); Ouyang et al. (2022). For mathematical reasoning, recent systems increasingly use verifiable rewards that score model outputs against known answers or programmatic checks. Systems such as DeepSeekMath Shao et al. (2024), DeepSeek-R1 DeepSeek-AI (2025), Kimi k1.5 Team et al. (2025), and Qwen-Math Yang et al. (2024) demonstrate the effectiveness of this approach. A common optimizer in this setting is Group Relative Policy Optimization (GRPO) Shao et al. (2024), which estimates relative advantages from groups of sampled completions without learning a separate value model.

The usefulness of a GRPO update, however, depends on reward variation within each sampled group. When every completion receives effectively the same shaped verifier reward, the reward derived advantages for that group vanish. The KL regularizer may remain active, but there is no reward derived policy gradient from the group. This creates a simple question: when groups with little or no reward variation become common, should training continue with the same objective, or should it change how the remaining reward information is used?

We study this question through Contrastive Self-Play (CSP). For groups with a sufficient shaped reward gap, CSP contrasts a higher reward completion with a lower reward completion and optimizes the resulting preference. Groups with zero shaped reward variance do not provide such a pair, so CSP does not recover supervision that is absent from the verifier. Instead, it changes how the remaining

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.

reward variation is converted into an update. Applying CSP throughout training is not a complete solution either. In our experiments, always on CSP can reach competitive peak accuracy but regress substantially later in training, which suggests that neither GRPO nor CSP is uniformly preferable throughout optimization.

We therefore introduce **Adaptive Contrastive Self-Play (A-CSP)**, which changes the optimization objective according to the observed reward regime. A-CSP monitors `zv_frac`, the fraction of rollout groups with effectively zero shaped reward variance, and uses hysteresis to switch between GRPO and CSP. The statistic is available from the same rewards already collected for GRPO and adds no separate model or prediction stage. Because the verifier rewards are shaped, `zv_frac` measures equality in the implemented reward rather than correctness diversity. Its numerical value is also task specific, and we use it as an operational training signal rather than as a sufficient rule for deciding which objective is better.

Across 156 valid training runs on GSM8K and a custom MATH-L1-3 pool, including 48 transfer runs across model and task cells, we find that the value of switching is strongly regime dependent. On Qwen2.5-1.5B GSM8K, A-CSP reaches similar peak exact match to GRPO, 0.730 versus 0.737, but retains substantially more of its peak performance by the end of training, 0.97 versus 0.90. On Qwen2.5-3B, A-CSP improves peak accuracy on both tasks and final accuracy on GSM8K, while final accuracy on MATH-L1-3 is essentially unchanged. Controlled decompositions show that masking KL on zero variance groups does not reproduce the A-CSP behavior, while changing which groups contribute to the update has a larger effect on the optimization dynamics in the flagship setting. At the same time, simple switching schedules can match or exceed A-CSP in peak accuracy. The results therefore point to a qualified conclusion: adaptive objective switching can help, but the prevalence of zero variance groups alone does not determine when switching is beneficial.

Our contributions are threefold. First, we introduce A-CSP, a simple method that switches between group relative and contrastive optimization using a statistic already available from rollout rewards. Second, we provide a controlled study of objective choice in RLVR, including GRPO decomposition, switching schedule controls, switching thresholds, and evaluation across models and reasoning tasks. Third, we identify clear boundary conditions for adaptive switching: A-CSP improves retention and performance in several settings, but its advantage is heterogeneous, simple schedules can be competitive, and zero reward variance is an informative training state rather than a sufficient objective selector.

2 Background

GRPO learns from relative rewards within a sampled group, while preference optimization learns from an ordered pair of responses. We review the parts of these objectives that are needed for our method.

Group Relative Policy Optimization. Given a prompt x, GRPO Shao et al. (2024) samples G completions $\{y_i\}_{i=1}^{G}$ from the current policy and assigns each completion a reward r_i. It forms relative advantages from the reward mean μ_r and standard deviation σ_r,

$$A_i = \frac{r_i - \mu_r}{\sigma_r}, \tag{1}$$

and uses these advantages in the policy update together with regularization toward a reference policy. Unlike PPO Schulman et al. (2017), GRPO does not require a learned value function. The sampled group provides the baseline used to construct relative advantages.

The property important to this work follows directly from the normalization. If $\sigma_r = 0$, then every completion in the group has zero advantage, so that group contributes no policy gradient from reward. Other terms in the objective, including KL regularization, may remain active. In our experiments, r_i is the implemented shaped verifier reward, so zero reward variance refers to equality of shaped rewards rather than equality of correctness labels.

Direct Preference Optimization. Direct Preference Optimization (DPO) Rafailov et al. (2023) learns from an ordered pair (y^+, y^-) by increasing the relative likelihood of the preferred response. Classical DPO uses a frozen reference policy π_{ref} and compares log likelihoods of the preferred

Table 1: **Shaped verifier rewards used during training.** The same reward values are used by GRPO, CSP, and the A-CSP controller.

Setting	Reward rule
GSM8K	Correct: 1.0, and otherwise $+0.1$ for a #### marker or any parseable number.
MATH-L1-3	Correct: 1.0, and otherwise $+0.1$ for a boxed extraction, $+0.1$ for a parseable number, plus $0.2(1 - \mathrm{Lev}/\mathrm{max_len})$ from normalized edit similarity. Incorrect responses therefore receive rewards in $[0, 0.4]$.

and dispreferred completions under the current policy relative to that reference. Online preference methods construct such pairs from responses sampled during training rather than from a fixed offline preference dataset Pang et al. (2024); Chen et al. (2024); Azar et al. (2024).

CSP builds on this pairwise view but uses verifier rewards from the current rollouts. When a group contains a sufficient reward gap, CSP selects a higher reward response and a lower reward response and applies a DPO style update. The exact objective differs from classical DPO: our CSP branch uses detached weights derived from the current policy and does not use a frozen reference policy. Section 3.2 defines this objective precisely.

Together, these two views expose the distinction used by A-CSP. GRPO converts variation across a rollout group into relative advantages, while CSP converts eligible reward differences into pairwise updates. Neither objective creates verifier information when the observed rewards provide no usable distinction. Section 3 describes how A-CSP switches between them.

3 METHOD

A-CSP uses the same rollout rewards in two different ways. GRPO forms relative advantages from reward variation within a group, while CSP forms a pair from the highest and lowest reward completions when their reward gap is sufficiently large. We first define the shaped rewards and the eligibility conditions used by both objectives, then describe CSP and the adaptive controller.

3.1 REWARD GROUPS AND ELIGIBILITY

For each prompt x, the current policy π_θ samples G completions $\{y_i\}_{i=1}^G$ and assigns shaped verifier rewards $r_i = r(y_i, x)$. The same raw reward values are used to compute GRPO advantages, determine CSP eligibility, and update the A-CSP controller. Table 1 gives the reward rules used in our experiments.

For a rollout group, define its reward standard deviation and reward range as

$$s(x) = \mathrm{std}(r_1, \ldots, r_G), \qquad d(x) = \max_i r_i - \min_i r_i. \tag{2}$$

We call a group **zero variance** when

$$s(x) < \epsilon_{\mathrm{zv}}, \qquad \epsilon_{\mathrm{zv}} = 10^{-8}. \tag{3}$$

A group is **GRPO informative** when $s(x) \geq \epsilon_{\mathrm{zv}}$ and **CSP eligible** when

$$d(x) > \epsilon_{\mathrm{csp}}, \qquad \epsilon_{\mathrm{csp}} = 0.01. \tag{4}$$

For a training step containing B prompt groups, A-CSP summarizes the prevalence of zero variance groups by

$$z_t = \texttt{zv_frac}_t = \frac{1}{B} \sum_{b=1}^{B} \mathbf{1}\left[s(x_b) < \epsilon_{\mathrm{zv}}\right]. \tag{5}$$

Thus, `zv_frac` measures equality in the implemented shaped reward rather than equality of correctness labels.

The relationship between GRPO informativeness and CSP eligibility depends on the reward function. On GSM8K, the smallest nonzero reward gap is 0.1, so every GRPO informative group is also CSP

eligible. A CSP pair can nevertheless contain two incorrect completions if their shaped rewards differ. On MATH-L1-3, the edit term can change in increments of $0.2/\max_\text{len}$. A group can therefore have nonzero reward variance while its reward range remains at most 0.01, making it informative for GRPO but ineligible for CSP. Because GSM8K and MATH-L1-3 use different shaped reward functions, raw values of `zv_frac` should not be compared directly across tasks.

3.2 Contrastive Self-Play

For each CSP eligible group, we select the completions at the two reward extremes,

$$y^+ = \arg\max_i r_i, \qquad y^- = \arg\min_i r_i. \tag{6}$$

The pair is retained only when

$$r(y^+, x) > r(y^-, x) + \epsilon_{\text{csp}}. \tag{7}$$

Ties at either reward extreme are resolved by the implementation. Groups that do not satisfy this condition contribute no CSP pair.

For a valid pair, let

$$\ell^+ = \log \pi_\theta(y^+ \mid x), \qquad \ell^- = \log \pi_\theta(y^- \mid x), \tag{8}$$

where $\ell^\pm$ is the total log probability of the completion tokens under π_θ, and define the detached pair weight

$$w = \sigma\left(-\beta\left[\text{stopgrad}(\ell^+) - \text{stopgrad}(\ell^-)\right]\right). \tag{9}$$

CSP averages the following objective over the valid pairs in the step:

$$\mathcal{L}_{\text{CSP}} = \frac{1}{n_{\text{pairs}}} \sum_{(x,y^+,y^-)} \left[-w\beta\ell^+ + w\beta\ell^- + \alpha\ \text{NLL}(y^+ \mid x) - \eta\,\mathcal{H}(\pi_\theta(\cdot \mid x))\right]. \tag{10}$$

with $\text{NLL}(y^+ \mid x) = -\ell^+/|y^+|$, where $|y^+|$ is the token length of the chosen completion. The first two terms form the pairwise update, the supervised term increases the likelihood of the higher reward completion, and the entropy term regularizes the current policy.

Unlike classical DPO, CSP does not use a frozen reference policy in this branch. The pair weight is computed from current policy log probabilities and detached before backpropagation. If a step contains no valid pair, CSP performs no parameter update. Algorithm 2 in Appendix G summarizes pair construction and the CSP update.

3.3 Adaptive Contrastive Self-Play

A-CSP selects between GRPO and CSP using `zv_frac`. Training begins in GRPO mode. Every 50 optimization steps, the controller evaluates `zv_frac` on the current rollout batch. If the statistic is strictly greater than θ_{up} at two consecutive controller checks, A-CSP switches to CSP. While in CSP mode, a single check strictly below θ_{down} returns training to GRPO. Unless otherwise stated, we use

$$\theta_{\text{up}} = 0.30, \qquad \theta_{\text{down}} = 0.20. \tag{11}$$

Algorithm 3 in Appendix G gives the complete controller procedure.

The two thresholds and asymmetric confirmation create hysteresis between the objectives. A mode change made after step t takes effect at step $t+1$. The controller uses only rewards already produced by the rollout batch, so choosing the optimization mode requires no additional model call.

Discrete controller thresholds. Each training step contains $B = 8$ prompt groups, so `zv_frac` can take only values in $\{0, 1/8, \ldots, 1\}$. With $\theta_{\text{up}} = 0.30$, the first attainable value satisfying the strict inequality is 0.375. With $\theta_{\text{down}} = 0.20$, returning to GRPO requires `zv_frac` ≤ 0.125. The nominal upper thresholds 0.27, 0.30, and 0.35 therefore share the same effective entry boundary. Appendix E reports the corresponding threshold analysis.

Table 2: **Main performance across models and tasks** (H100 rerun, three seeds, dedicated workers). Peak EM and Final EM are means across seeds on the held out evaluation subset of 100 examples, with sample standard deviation (n=3, $ddof$=1). Retention is the mean across seeds of Final EM divided by Peak EM.

Model	Task	GRPO			A-CSP		
		Peak EM	Final EM	Ret.	Peak EM	Final EM	Ret.
Qwen2.5-1.5B	GSM8K	.737 ±.015	.667 ±.067	.90	.730 ±.020	.707 ±.029	.97
Qwen2.5-1.5B	MATH-L1-3	.363 ±.021	.330 ±.010	.91	.347 ±.031	.280 ±.017	.81
Qwen2.5-3B	GSM8K	.867 ±.015	.533 ±.437	.62	.880 ±.010	.747 ±.057	.85
Qwen2.5-3B	MATH-L1-3	.390 ±.026	.377 ±.045	.97	.510 ±.046	.377 ±.127	.74
Llama-3.2-3B	GSM8K	.803 ±.021	.000 ±.000	.00	.837 ±.015	.707 ±.084	.84
Llama-3.2-3B	MATH-L1-3	.623 ±.055	.000 ±.000	.00	.630 ±.030	.000 ±.000	.00

4 Experiments

4.1 Experimental Setup

We evaluate A-CSP against GRPO on `Qwen2.5-1.5B-Instruct` and `Qwen2.5-3B-Instruct` Yang et al. (2024), using LoRA ($r = 16$, $\alpha = 32$ on attention projections) Hu et al. (2022). We train on GSM8K Cobbe et al. (2021) and **MATH-L1-3**, a custom pool drawn from Hendrycks MATH Levels 1–3 Hendrycks et al. (2021). MATH-L1-3 is not the canonical MATH500 benchmark. Evaluation uses exact match on normalized answers, with the GSM8K numeric verifier and MATH style boxed extraction for MATH-L1-3. These evaluation labels are distinct from the shaped training rewards in Table 1.

Training runs for 1,000 steps with eight prompts per step and $G = 8$ completions per prompt. The primary A-CSP configuration uses entropy coefficient $\eta = 0.001$. All remaining optimization and controller settings are given in Table N.12.

The reported H100 suite contains 156 valid training runs, including 48 transfer runs across eight model and task cells. Each Ray worker is allocated one H100 80GB GPU on AWS `p5.48xlarge` nodes. Unless otherwise stated, results use seeds $\{42, 123, 7\}$. The main comparisons use dedicated execution. MPS packed SmolLM2 runs are reported separately as execution diagnostics in Appendix B and are excluded from method level timing comparisons in Appendix I.

We report three trajectory metrics. **Peak EM** is the maximum exact match over scheduled evaluations. **Final EM** is exact match at the final scheduled evaluation at step 1,000. **Retention** is the mean across seeds of the per seed ratio of Final EM to Peak EM. Reported ± values for Peak EM and Final EM are sample standard deviations across three seeds ($ddof$=1).

Each run uses 500 training prompts and a fixed 100 example evaluation subset. Training samples eight distinct prompts per step for 1,000 steps, giving about 16 expected exposures per training example. Evaluation is performed at step 1 and every 50 steps with greedy decoding. Peak EM is the retrospective maximum over these evaluations, while Final EM is the final scheduled evaluation on the same subset and is not a separate locked test. GSM8K evaluation uses the HuggingFace test split. MATH-L1-3 uses a disjoint slice of the seed shuffled custom pool. Methods compared at the same seed share the same split. Full protocol details are given in Appendix F.

4.2 A-CSP has regime dependent effects

Table 2 reports dedicated path comparisons on six model and task cells. A-CSP changes late training behavior more consistently than it changes peak accuracy, and its effect is not uniform across models and tasks.

On **Qwen2.5-1.5B GSM8K**, our flagship setting, A-CSP and GRPO reach similar Peak EM: 0.730 and 0.737, respectively. Their trajectories differ more at the end of training. A-CSP reaches Final EM 0.707 with retention 0.97, compared with 0.667 and 0.90 for GRPO. The main effect in this setting is therefore improved retention rather than improved peak accuracy.

Table 3: **GRPO decomposition on Qwen2.5-1.5B GSM8K** (three paired seeds). Rollout cost is mean generation time relative to standard GRPO ($= 1.0\times$). Retention is the mean across seeds of Final EM divided by Peak EM.

Method	Intervention	Peak EM	Final EM	Ret.	Rollout cost
GRPO	baseline	0.737	0.667	0.90	$1.0\times$
GRPO	masked KL on zero variance groups	0.737	0.700	0.95	$1.02\times$
GRPO	informative groups only	0.733	0.683	0.93	$1.04\times$
GRPO	DAPO style dynamic sampling	0.727	0.693	0.95	$2.52\times$
A-CSP	primary controller	0.730	0.707	0.97	$0.99\times$

Masking KL leaves peak EM unchanged, while filtering to informative groups shifts peak EM slightly toward the A-CSP value. Dynamic sampling increases rollout cost by about $2.5\times$ without improving peak EM in this setting.

The difference is larger on **Qwen2.5-3B GSM8K**. Peak EM remains similar, at 0.880 for A-CSP and 0.867 for GRPO, but Final EM increases from 0.533 to 0.747 and retention from 0.62 to 0.85. GRPO also has substantially larger variation across seeds at the final evaluation. On **Qwen2.5-3B MATH-L1-3**, the pattern is different: A-CSP raises Peak EM from 0.390 to 0.510, but both methods end at mean Final EM 0.377. The higher peak is therefore not retained through the end of training.

The remaining cells show that these gains are not universal. On **Qwen2.5-1.5B MATH-L1-3**, A-CSP is lower than GRPO on both Peak EM (0.347 versus 0.363) and Final EM (0.280 versus 0.330). On **Llama-3.2-3B GSM8K**, GRPO reaches Final EM 0 on all three seeds, whereas A-CSP ends at 0.707 with retention 0.84. On **Llama-3.2-3B MATH-L1-3**, however, both methods end at Final EM 0 despite reaching nonzero peaks.

The transfer cells use one shared configuration rather than tuning hyperparameters for each model family. The results therefore support a bounded conclusion: A-CSP can substantially improve late training behavior in some settings, but it does not dominate GRPO across models, tasks, or evaluation endpoints. SmolLM2 runs use different execution paths and are reported only as diagnostics in Appendix B.

4.3 What changes the GRPO dynamics?

Masking KL on zero variance groups does not reproduce the change in peak behavior observed with A-CSP on Qwen2.5-1.5B GSM8K in Table 3. Standard GRPO reaches Peak EM 0.737, and masking KL on zero variance groups leaves the mean unchanged at 0.737. The paired difference is $+0.000 \pm 0.017$ across three seeds.

Changing which groups contribute to GRPO has a small but different effect. Restricting updates to informative groups moves Peak EM to 0.733 (paired difference -0.003 ± 0.012), numerically toward the A-CSP value of 0.730. The intervention does not reproduce A-CSP retention, reaching 0.93 compared with 0.97 for A-CSP. Masked KL reaches retention 0.95 and Final EM 0.700, so neither modification reproduces the full A-CSP trajectory.

DAPO style dynamic sampling provides a separate comparison in which uninformative groups are resampled rather than filtered from the update. It reaches Peak EM 0.727 and Final EM 0.693, while increasing rollout cost to $2.52\times$ standard GRPO. In this setting, the additional sampling cost does not improve peak accuracy.

These comparisons support a limited interpretation. Changing batch composition affects the optimization outcome in ways that KL masking alone does not, but the decomposition does not establish informative group filtering as the cause of A-CSP's retention behavior.

4.4 Does adaptive timing matter?

Table 4 compares fixed switching schedules on GSM8K. Adaptive timing is not required to obtain strong peak performance. On **Qwen2.5-1.5B**, a random schedule using only 31 CSP steps reaches Peak EM 0.750 and Final EM 0.733, compared with 0.730 and 0.707 for A-CSP. The periodic

Table 4: **Switching schedule controls on GSM8K** (three seeds, H100 rerun). CSP steps is the mean number of CSP optimizer steps across seeds. Only the shuffled schedule approximately matches A-CSP in CSP exposure within each model size. Peak EM and Final EM are means across seeds with sample SD ($ddof=1$). Retention is the mean across seeds of Final EM divided by Peak EM.

Model	Schedule	CSP steps	Peak EM	Final EM	Ret.
Qwen2.5-1.5B	A-CSP	283	.730 ±.020	.707 ±.029	.97
Qwen2.5-1.5B	periodic	38	.743 ±.006	.687 ±.020	.92
Qwen2.5-1.5B	random	31	.750 ±.010	.733 ±.025	.98
Qwen2.5-1.5B	shuffled	283	.740 ±.012	.663 ±.040	.90
Qwen2.5-3B	A-CSP	900	.880 ±.010	.747 ±.057	.85
Qwen2.5-3B	periodic	29	.857 ±.015	.347 ±.410	.40
Qwen2.5-3B	random	23	.853 ±.030	.533 ±.410	.62
Qwen2.5-3B	shuffled	901	.873 ±.030	.670 ±.120	.77

schedule similarly reaches a higher mean Peak EM of 0.743 with only 38 CSP steps. Thus, the controller does not provide the highest peak or final mean among the schedules tested in the flagship setting.

The pattern changes at **Qwen2.5-3B**. A-CSP reaches Peak EM 0.880 and Final EM 0.747. Periodic and random switching reach Peak EM 0.857 and 0.853, respectively, but their Final EM falls to 0.347 and 0.533 with large variation across seeds. A schedule that works well at 1.5B therefore need not preserve the same behavior at 3B.

The shuffled schedule provides the closest comparison of timing at approximately matched CSP exposure. At 1.5B, both A-CSP and shuffled switching use 283 CSP steps. Shuffled switching reaches a slightly higher mean Peak EM (0.740 versus 0.730), but a lower Final EM (0.663 versus 0.707). At 3B, shuffled switching uses 901 CSP steps compared with 900 for A-CSP and reaches Peak EM 0.873 versus 0.880, while Final EM is 0.670 versus 0.747. Matched CSP exposure therefore does not give A-CSP a consistent advantage in peak accuracy, but A-CSP has higher mean Final EM in both model sizes.

Periodic and random schedules use far fewer CSP steps and therefore vary both timing and total contrastive exposure. Their strong 1.5B results also show that CSP exposure alone does not determine peak performance. The schedule controls are consistent with timing affecting late training behavior, but they do not establish `zv_frac` as an optimal rule for choosing when to switch.

5 Analysis

The main experiments establish that A-CSP can change late training behavior, but that its benefit is heterogeneous and cannot be attributed to adaptive timing alone. We now examine three supporting questions: how controller exposure relates to performance, how the CSP configuration affects late training, and whether declining lexical diversity explains regression.

5.1 Controller exposure alone does not determine benefit

Appendix Table H.9 reports GRPO mode `zv_frac` and CSP exposure by cell. A-CSP spends very different fractions of training in CSP across model and task settings. Consistent with the reward definitions in Section 3.1, we interpret `zv_frac` only within a task because GSM8K and MATH-L1-3 use different shaped reward functions. Neither the observed zero variance fraction nor total CSP exposure alone orders the performance gains.

For example, Qwen2.5-3B GSM8K spends about 900 of 1,000 steps in CSP and shows a large improvement in Final EM, whereas Qwen2.5-3B MATH-L1-3 spends about half of training in CSP but ends at the same mean Final EM as GRPO. Llama-3.2-3B MATH-L1-3 receives substantial CSP exposure and still ends at Final EM 0 under both methods. These cells reinforce the conclusion from Section 4.2: controller activity is a description of the training trajectory, not a sufficient predictor of benefit.

Total exposure also does not isolate repeated adaptivity. On Qwen2.5-3B GSM8K, A-CSP remains in CSP for roughly 90% of training. We did not evaluate a control consisting of one initial GRPO phase followed by a single CSP block with matched exposure. This setting therefore cannot distinguish the effect of repeated adaptive switching from that of spending most of training under CSP.

SmolLM2 provides a further diagnostic boundary case, including a MATH-L1-3 run with zero CSP exposure. Because execution paths differ, we leave its interpretation to Appendix B.

5.2 CSP CONFIGURATION AFFECTS LATE TRAINING BEHAVIOR

The CSP schedule and objective composition affect peak and final performance differently. At the matched entropy coefficient $\eta = 0.01$, adaptive A-CSP reaches Final EM 0.677, whereas E-CSP, which enters CSP after step 50 and remains there, reaches 0.473 in Appendix Table C.6. Always on CSP reaches a similar Final EM of 0.483. These comparisons are consistent with prolonged CSP exposure contributing to late regression, but they do not identify the mechanism responsible for that regression.

Appendix Table D.7 ablates CSP objective terms under the same adaptive controller. Several partial CSP objectives reach higher Peak EM than the full A-CSP objective, while the full configuration has the highest retention among the adaptive variants tested. The results therefore do not identify a single CSP term that accounts for A-CSP behavior. The effect depends on how the pairwise, supervised, and entropy terms interact over the training trajectory.

5.3 LEXICAL DIVERSITY DOES NOT EXPLAIN LATE REGRESSION

Appendix Figure J.1 plots offline lexical diversity on saved checkpoints. Direct lexical diversity probes do not support a monotonic homogenization account of late CSP regression. The measured diversity remains flat or increases in trajectories where exact match can decline. Reduced lexical diversity is therefore not sufficient to explain the observed regression.

These probes are descriptive rather than mechanistic. They reject one simple explanation but do not identify an alternative cause. The source of late regression remains unresolved.

6 RELATED WORK

Preference optimization from model generated reasoning. Iterative Reasoning Preference Optimization Pang et al. (2024) is closest to CSP: both construct preference pairs from sampled reasoning and combine pairwise and supervised objectives. IRPO trains iteratively from binary correctness labels, whereas CSP forms one pair from shaped reward extremes in the current rollout and updates the same policy without a frozen reference model. SPIN Chen et al. (2024), Self-Rewarding Language Models Yuan et al. (2024), and STaR Zelikman et al. (2022) also learn from model generated outputs, but use different sources of preference or supervised signal.

Low reward variance in GRPO. DAPO Yu et al. (2025) resamples after filtering zero variance groups, while NGRPO Nan et al. (2025) and GVPO Zhang et al. (2025) modify the policy optimization rule. A-CSP instead leaves GRPO unchanged while active and switches to a separate pairwise objective when zero variance groups become common. It does not create supervision for groups whose shaped rewards contain no usable distinction.

Adapting data and rewards. VCRL Jiang et al. (2025) uses reward variance to control the sampling curriculum, and TA-GRPO Le et al. (2026) pools semantically equivalent problem variants when computing advantages. Math-Shepherd Wang et al. (2024) instead provides denser process supervision. These methods change the data or reward signal. A-CSP changes the optimization objective applied to a fixed rollout distribution.

7 LIMITATIONS

Scope and evaluation. Our experiments study mathematical RLVR with shaped deterministic verifier rewards, model sizes up to 3B parameters, and two reasoning tasks. MATH-L1-3 is a custom pool

rather than the canonical MATH500 benchmark, and all reported exact match values are measured on seed dependent held out subsets of 100 examples. We did not run a separate locked final test evaluation. The results should therefore be interpreted as evidence about optimization behavior under this training and evaluation protocol rather than as estimates of full benchmark performance or general guarantees for other domains, verifier designs, or model scales.

Controller scope. A-CSP uses `zv_frac`, which measures equality of the implemented shaped reward and is not directly comparable across tasks with different reward functions. The controller uses fixed empirical thresholds and asymmetric hysteresis rather than a learned or task calibrated switching rule. Our schedule controls also show that simple fixed schedules can match or exceed A-CSP on peak performance in some settings. Only the shuffled schedule approximately matches total CSP exposure, and matched exposure does not imply matched block structure. In particular, Qwen2.5-3B GSM8K spends about 90% of training in CSP, and we did not evaluate a control with one initial GRPO phase followed by a single CSP block of comparable length. The experiments therefore do not establish that repeated adaptive switching is necessary in this regime or that `zv_frac` defines an optimal switching policy.

Statistical precision and mechanistic interpretation. Primary comparisons use three seeds and 100 example evaluation subsets. Small differences in Peak EM, such as those between GRPO and A-CSP on Qwen2.5-1.5B GSM8K, should therefore not be interpreted as reliable differences in underlying model quality, and variation across seeds also includes variation in the held out evaluation subset. The decomposition and schedule controls constrain several explanations but do not identify a complete mechanism for A-CSP's behavior. Our diversity analysis measures surface lexical statistics on saved checkpoints and does not measure semantic diversity or the diversity of pairs used inside CSP updates. The absence of lexical homogenization therefore rules out only one simple explanation for late regression, and the source of that regression remains unresolved.

8 Conclusion

We studied whether reinforcement learning with verifiable rewards should use the same optimization objective throughout training. Adaptive Contrastive Self-Play (A-CSP) uses the prevalence of zero variance shaped reward groups to switch between GRPO and a contrastive objective that learns from eligible reward differences. Across models and tasks, the effect of switching is heterogeneous: A-CSP improves late training behavior in several settings but does not uniformly improve peak performance or outperform GRPO. Our decomposition does not support KL masking alone as an explanation for the observed behavior, while the schedule controls show that simple switching rules can match or exceed A-CSP on peak accuracy. Thus, `zv_frac` is useful as an operational signal of training state, but it is not sufficient to determine the best optimization objective.

The broader lesson is that the objective used in RLVR need not remain fixed throughout training, but deciding when to change it requires more information than zero reward variance alone.

AI Use Statement

Generative AI tools were used for limited research support, including feedback on experimental design, software implementation and data formatting, literature review support, interpretation of experimental results, and editorial revision of the manuscript. All AI assisted outputs were reviewed and verified by the authors. The authors designed the study, implemented and tested the core methods, conducted the experiments and analyses, and take full responsibility for the final text, claims, code, results, and artifacts. Generative AI was not used to generate synthetic datasets or mathematical proofs.

Reproducibility Statement

We provide the method definitions, reward functions, training and evaluation protocols, hyperparameters, model and dataset configurations, random seeds, controller settings, ablations, and statistical aggregation procedures needed to interpret the reported results. The appendices document additional execution details, timing methodology, robustness analyses, and known experimental limitations,

including experiments excluded from scientific analysis. The code used to run the experiments and analyses will be released upon acceptance.

REFERENCES

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455. PMLR, 2024.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning convertsweak language models to strong language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 4302–4310, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Guochao Jiang, Wenfeng Feng, Guofeng Quan, Chuzhan Hao, Yuewei Zhang, Guohua Liu, and Hao Wang. Vcrl: Variance-based curriculum reinforcement learning for large language models. *arXiv preprint arXiv:2509.19803*, 2025.

Khiem Le, Youssef Mroueh, Phuc Nguyen, Chi-Heng Lin, Shangqian Gao, Ting Hua, and Nitesh V Chawla. Transform-augmented grpo improves pass@ k. *arXiv preprint arXiv:2601.22478*, 2026.

Gongrui Nan, Siye Chen, Jing Huang, Mengyu Lu, Dexun Wang, Chunmei Xie, Weiqi Xiong, Xianzhou Zeng, Qixuan Zhou, Yadong Li, et al. Ngrpo: Negative-enhanced group relative policy optimization. *arXiv preprint arXiv:2509.18851*, 2025.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. Iterative reasoning preference optimization. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: your language model is secretly a reward model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

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.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1.5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, 2024.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Neural Information Processing Systems*, 2025. doi: 10.48550/arXiv.2503.14476.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: self-taught reasoner bootstrapping reasoning with reasoning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Kaichen Zhang, Yuzhong Hong, Junwei Bao, Hongfei Jiang, Yang Song, Dingqian Hong, and Hui Xiong. Gvpo: Group variance policy optimization for large language model post-training. *arXiv preprint arXiv:2504.19599*, 2025.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019. URL https://arxiv.org/abs/1909.08593.

APPENDIX

A VR-SPGRPO: VARIANCE-ROBUST SELF-PLAY GRPO

VR-SPGRPO explores a different response to zero variance rollout groups from A-CSP. Rather than changing the optimization objective, it asks the current policy to rewrite a prompt when the original rollout group has effectively equal shaped rewards. The rewritten prompts are intended to produce alternative rollout groups with more useful reward variation. Whether rewriting actually restores such variation is a design hypothesis. The experiments in this paper do not establish it as the operative mechanism. As throughout the paper, zero shaped reward variance does not imply that all completions are correct or all are incorrect.

VR-SPGRPO adds three main mechanisms to standard GRPO. First, a *memory baseline* maintains a per prompt exponential moving average of past rewards, with update rate $\alpha = 0.05$. When rewritten responses are generated for a zero variance group, their advantages are measured relative to this stored baseline rather than normalized within the original degenerate group. Second, a *zero variance curriculum* reduces the sampling weight of prompts that repeatedly produce zero shaped reward variance, using multiplicative factor $1 - \tau$ with $\tau = 0.1$. Third, a *Self-Play rewriter* generates between one and three prompt variants using rewrite goals drawn from $\{\texttt{clarify}, \texttt{adversarial}, \texttt{paraphrase}, \texttt{decompose}\}$. For $K = 3$, the $G = 8$ rollout budget is distributed across the rewritten variants.

The broader prototype also supports an optional neural verifier that combines rule based correctness with a learned confidence score. This component is separate from the primary A-CSP method and is not needed to define the rewriting procedure studied here.

Algorithm 1 summarizes the update. For each sampled prompt, the trainer first generates a standard rollout group and measures its shaped reward variance. Groups with nonzero variance use the usual GRPO advantages. For zero variance groups, the trainer generates rewritten variants, samples new completions, and computes their advantages relative to the stored memory baseline. The curriculum state and memory are then updated, and the actor receives one clipped GRPO update on the resulting training tuples.

Algorithm 1: VR-SPGRPO update

Input: dataset $\mathcal{D}$, policy π_θ, memory $\mathcal{M}$, curriculum $\mathcal{C}$, group size G, rewrite count K

Sample prompt batch $\mathcal{B}$ from $\mathcal{D}$ according to curriculum weights $\mathcal{C}$;
$\mathcal{U} \leftarrow \emptyset$;
foreach $x \in \mathcal{B}$ **do**
  Sample G completions from $\pi_\theta(\cdot|x)$ and compute rewards $\{r_i\}_{i=1}^{G}$;
  $\mu_r \leftarrow \mathrm{mean}(\{r_i\})$;
  $\sigma_r \leftarrow \mathrm{std}(\{r_i\})$;
  if $\sigma_r < \epsilon_{\mathrm{zv}}$ **then**
    Generate K rewritten prompts $\{x^{(k)}\}_{k=1}^{K}$;
    for $k = 1, \ldots, K$ **do**
      Sample $\lfloor G/K \rfloor$ completions from $\pi_\theta(\cdot|x^{(k)})$;
      foreach *completion* $y_j^{(k)}$ *with reward* $r_j^{(k)}$ **do**
        $A_j^{(k)} \leftarrow r_j^{(k)} - \mathcal{M}(x)$;
        Add $(x^{(k)}, y_j^{(k)}, A_j^{(k)})$ to $\mathcal{U}$;
    Decrease the curriculum weight of x;
  else
    foreach *completion* y_i **do**
      $A_i \leftarrow (r_i - \mu_r)/\sigma_r$;
      Add (x, y_i, A_i) to $\mathcal{U}$;
    Restore the curriculum weight of x;
  Update memory $\mathcal{M}(x)$ toward μ_r;
Perform one clipped GRPO update using $\mathcal{U}$;

We report VR-SPGRPO only as an appendix pilot. On `SmolLM2-1.7B` GSM8K, it reaches mean Final EM 0.475 with full retention, while GRPO reaches a higher peak but shows substantial late regression, including one seed with 47% retention. These runs are outside the main H100 A-CSP matrix and do not establish a benefit that transfers to Qwen models. They also do not establish that prompt difficulty causes high `zv_frac`, or that rewriting improves training by restoring reward variation.

The role of this pilot is therefore limited. It tests prompt rewriting as an alternative response to degenerate rollout groups, whereas A-CSP keeps the prompts fixed and changes the optimization objective. No primary claim in the paper depends on VR-SPGRPO.

B SmolLM2 Execution Diagnostics

Table B.5: **SmolLM2 execution path diagnostics** (three seeds). GRPO uses dedicated workers. A-CSP uses MPS packed orchestration. These rows are not interpreted as clean treatment effect estimates.

Task	GRPO (dedicated)			A-CSP (packed)			
	Peak EM	Final EM	Ret.	Peak EM	Final EM	Ret.	CSP steps
GSM8K	.503 $\pm$.015	.473 $\pm$.050	.94	.527 $\pm$.025	.470 $\pm$.036	.89	233
MATH-L1-3	.233 $\pm$.012	.167 $\pm$.025	.72	.387 $\pm$.047	.310 $\pm$.036	.80	0

The MATH-L1-3 A-CSP runs execute zero CSP steps, so their numerical difference from GRPO cannot be attributed to adaptive switching. GRPO uses dedicated execution, whereas A-CSP uses MPS packing with multiple runs per GPU. These execution paths differ from the dedicated Qwen and Llama comparisons.

The SmolLM2 runs use different execution paths for GRPO and A-CSP and are therefore reported only as diagnostics. In particular, the MATH-L1-3 A-CSP runs execute zero CSP steps, so their numerical difference from GRPO cannot be attributed to adaptive objective switching.

C Entropy and Locked CSP

The matched $\eta = 0.01$ comparison separates adaptive A-CSP from E-CSP, which enters CSP after step 50 and remains there. As discussed in Section 5.2, the difference in Final EM is consistent with prolonged CSP exposure contributing to late regression, but it does not identify the mechanism responsible.

D CSP Objective Ablation

These configurations share the A-CSP controller and differ only in the CSP objective terms enabled. Several partial objectives attain higher Peak EM than the full objective, while the full configuration has the highest retention among the adaptive variants tested. The ablation therefore does not identify a single objective term as sufficient to explain A-CSP behavior.

Table C.6: **Entropy and locked CSP on Qwen2.5-1.5B GSM8K** (H100 rerun, three seeds, sample SD, $ddof=1$). Adaptive rows use the A-CSP controller. E-CSP fixes $\eta = 0.01$ and enters CSP after step 50, whereas always on CSP uses CSP from the beginning of training. Retention is the mean across seeds of Final EM divided by Peak EM.

Condition	Entropy η	Peak EM	Final EM	Ret.
A-CSP (primary)	0.001	0.730 $\pm$0.020	0.707 $\pm$0.029	0.97
A-CSP	0.0	0.753 $\pm$0.012	0.653 $\pm$0.091	0.87
A-CSP	0.01	0.747 $\pm$0.032	0.677 $\pm$0.084	0.90
E-CSP (lock CSP)	0.01	0.717 $\pm$0.015	0.473 $\pm$0.080	0.66
Always on CSP	0.01	0.740 $\pm$0.015	0.483 $\pm$0.050	0.66
GRPO	n/a	0.737 $\pm$0.015	0.667 $\pm$0.067	0.90

Table D.7: **CSP objective ablation on Qwen2.5-1.5B GSM8K** (three seeds). Adaptive rows use the same A-CSP controller as the primary method. Each row enables only the CSP objective terms indicated, using the primary coefficient for each enabled term. When entropy is enabled, $\eta = 0.001$.

Configuration	Pairwise	SFT	Entropy	Peak EM	Final EM	Ret.
Pairwise only	✓			0.750	0.683	0.91
SFT only		✓		0.740	0.457	0.62
Pairwise + SFT	✓	✓		0.757	0.693	0.92
Pairwise + entropy	✓		✓	0.753	0.717	0.95
SFT + entropy		✓	✓	0.753	0.250	0.33
A-CSP (full terms)	✓	✓	✓	0.730	0.707	0.97

No single component alone reproduces full A-CSP retention. Several partial objectives reach higher peak EM, while the full objective gives the strongest retention among these adaptive configurations.

E Threshold Settings

Table E.8: **Nominal versus effective CSP entry thresholds** on Qwen2.5-1.5B GSM8K (B=8 prompt groups per step, θ_{down}=0.20). Each setting is evaluated with three independent seeds and need not match the primary A-CSP run at the same nominal θ_{up}. Discrete `zv_frac` implies the first attainable crossing above θ_{up}. Return to GRPO requires `zv_frac`<0.20, so only 0 and 0.125 qualify.

Nominal θ_{up}	Effective enter boundary	Mean Peak EM
0.27	0.375	0.727
0.30	0.375	0.733
0.35	0.375	0.737
0.40	0.500	0.750

Nominal settings 0.27–0.35 share one effective enter boundary. Differences reflect run to run stochastic variation rather than continuous threshold sensitivity.

Because each step contains $B = 8$ prompt groups, the controller observes `zv_frac` only on the discrete grid $\{0, 1/8, \ldots, 1\}$. Consequently, several nominal values of θ_{up} correspond to the same effective entry boundary. The threshold sweep should therefore be interpreted as a comparison of effective controller boundaries rather than as a continuous sensitivity analysis.

F Evaluation Protocol

Section 4.1 gives the full evaluation protocol. For completeness, GSM8K evaluation uses a fixed 100 example subset of the HuggingFace `test` split within each run. MATH-L1-3 uses indices $[500 : 600]$

of the seed shuffled pool and is disjoint from the training slice within that run. Evaluation uses exact match on normalized final answers rather than the shaped rewards used for training.

Peak EM is the maximum over scheduled trajectory evaluations and is used only as a retrospective summary. Final EM is measured at the final scheduled evaluation on the same 100 example subset. Retention is the mean across seeds of the per seed ratio of Final EM to Peak EM. No separate locked final test evaluation was run.

G CSP AND A-CSP PROCEDURES

Algorithm 2: Contrastive Self-Play update

Input: rollout batch $\mathcal{R}$, policy π_θ, β, α, η, CSP gap ϵ_{csp}

$\mathcal{P} \leftarrow \emptyset$;
foreach *prompt group* $(x, \{y_i, r_i\}_{i=1}^{G}) \in \mathcal{R}$ **do**
  $y^+ \leftarrow \arg\max_i r_i$;
  $y^- \leftarrow \arg\min_i r_i$;
  if $r(y^+, x) > r(y^-, x) + \epsilon_{\text{csp}}$ **then**
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{(x, y^+, y^-)\}$;
if $|\mathcal{P}| = 0$ **then**
  return
Compute $\mathcal{L}_{\text{CSP}}$ over $\mathcal{P}$ using Section 3.2;
Update θ with one optimizer step on $\mathcal{L}_{\text{CSP}}$;

Algorithm 3: Adaptive Contrastive Self-Play

Input: dataset $\mathcal{D}$, policy π_θ, prompt batch size B, group size G, controller period $m = 50$, thresholds $\theta_{\text{up}}, \theta_{\text{down}}$

mode $\leftarrow$ GRPO;
$c \leftarrow 0$;
for $t = 1, \ldots, T$ **do**
  Sample B prompts from $\mathcal{D}$ and generate G completions per prompt to obtain rollout batch $\mathcal{R}_t$;
  if mode $=$ GRPO **then**
    Perform one GRPO update using $\mathcal{R}_t$;
  else
    Perform one CSP update using Algorithm 2 on $\mathcal{R}_t$;
  if $t \bmod m = 0$ **then**
    Compute $z_t = \texttt{zv_frac}_t$ from $\mathcal{R}_t$;
    if mode $=$ GRPO **then**
      if $z_t > \theta_{\text{up}}$ **then**
        $c \leftarrow c + 1$;
      else
        $c \leftarrow 0$;
      if $c \geq 2$ **then**
        mode $\leftarrow$ CSP;
        $c \leftarrow 0$;
    else
      if $z_t < \theta_{\text{down}}$ **then**
        mode $\leftarrow$ GRPO;
        $c \leftarrow 0$;

Table H.9: **Controller behavior on A-CSP cells** (three seeds). GRPO mean zero variance fraction is the mean across seeds of each run's mean `zv_frac` over GRPO mode steps. CSP steps and CSP fraction are means across the three seeds from training logs. Raw `zv_frac` values should be interpreted within task because GSM8K and MATH-L1-3 use different shaped reward functions.

Cell	GRPO mean `zv_frac`	CSP steps	CSP fraction
Qwen2.5-1.5B GSM8K	0.33	283	0.28
Qwen2.5-1.5B MATH-L1-3	0.24	100	0.10
Qwen2.5-3B GSM8K	0.63	900	0.90
Qwen2.5-3B MATH-L1-3	0.54	517	0.52
SmolLM2-1.7B GSM8K	0.26	233	0.23
SmolLM2-1.7B MATH-L1-3	0.08	0	0.00
Llama-3.2-3B GSM8K	0.62	883	0.88
Llama-3.2-3B MATH-L1-3	0.63	667	0.67

SmolLM2-1.7B MATH-L1-3 records zero CSP steps in this rerun. See Appendix B.

H Controller Exposure

The table reports how often A-CSP enters the contrastive objective across model and task settings. Raw `zv_frac` values should be interpreted within task because the shaped reward functions differ between GSM8K and MATH-L1-3. As shown in Section 5.1, neither `zv_frac` nor total CSP exposure alone determines the observed performance difference.

I Dedicated Execution Timing

Table I.10: **Dedicated execution timing on Qwen2.5-1.5B GSM8K** (mean seconds per training step, three seeds). Only dedicated execution runs are used for method timing comparisons. Component means are rounded independently and need not sum exactly to the reported mean total.

Method	Generation	Update	Evaluation	Total	Rel.
GRPO	26.4	9.0	1.9	37.2	1.00×
A-CSP (η=0.001)	26.2	6.9	1.8	34.9	0.94×

Method timing comparisons use dedicated execution runs only. MPS packed runs characterize infrastructure throughput but are excluded from comparisons between training methods, and invalid fork continuations are excluded from all aggregates. The reported totals include generation, optimization, and evaluation time.

On dedicated `Qwen2.5-1.5B` GSM8K runs, primary A-CSP uses approximately $0.94\times$ the mean end to end time of GRPO. This difference is specific to the implementation and execution profile considered here, and we do not assign it to a single causal mechanism.

J Offline Diversity Trajectories

These measurements are computed offline from saved checkpoints and use surface lexical statistics. They do not measure semantic diversity or the diversity of response pairs used inside CSP updates. The probes therefore reject one simple account of late regression without identifying an alternative mechanism.

K Interpretation Criteria

Before inspecting the H100 mechanistic outcomes, we documented interpretation criteria covering masked KL, informative group filtering, DAPO style dynamic sampling, the planned fork experiment, switching schedules, adaptive triggering, CSP objective components, and lexical diversity. The corresponding scientific conclusions are reported in Sections 4.3, 4.4, and 5.

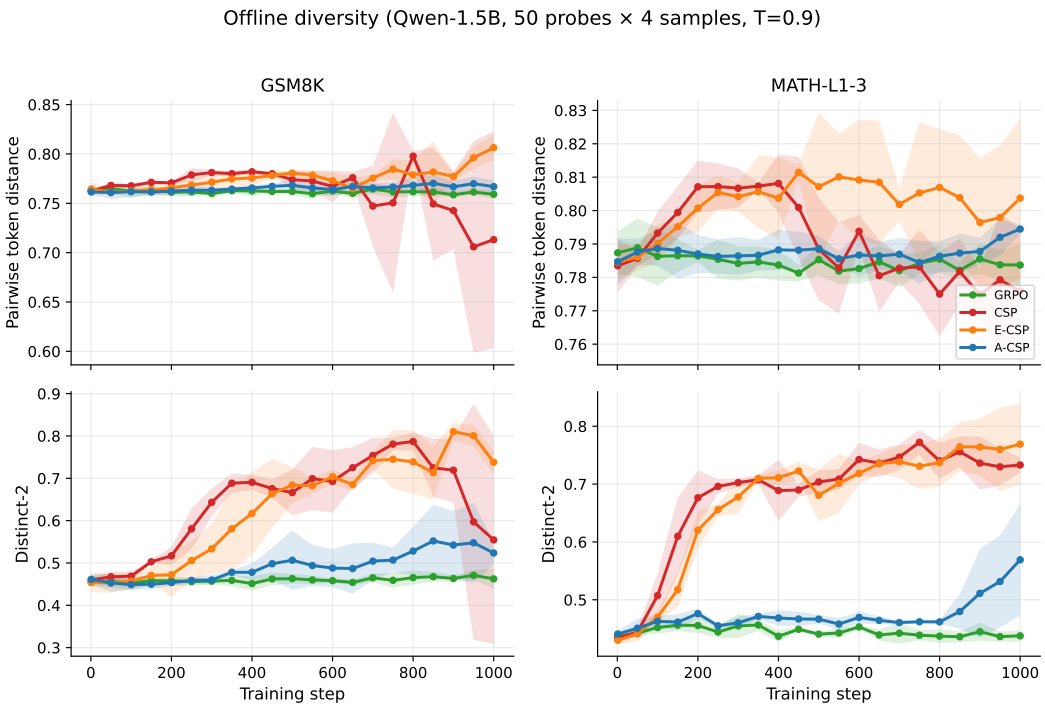

Figure J.1: **Offline lexical diversity probes** on `Qwen2.5-1.5B` checkpoints. `distinct_2` remains flat or increases along primary trajectories, which does not support lexical homogenization as an explanation for late regression.

Fork monotonicity criterion, not evaluated. The intended checkpoint stratified experiment was implemented incorrectly: the nominal `zv_frac` bands did not select different source checkpoints. We therefore exclude the resulting runs from all scientific analysis and draw no conclusion from this criterion.

The remaining definitions and their revision history are maintained in the project repository together with the cost taxonomy. The archival tag is a retrospective snapshot and should not be interpreted as the timestamp of every criterion revision.

L Controller Behavior on Qwen2.5-1.5B GSM8K and MATH-L1-3

On `Qwen2.5-1.5B` GSM8K, `zv_frac` frequently reaches the controller entry region, with median approximately 0.27 and excursions above 0.30. Because the statistic is computed from shaped rewards, these values do not imply that a group contains only correct or only incorrect responses.

MATH-L1-3 contains 5,964 problems from Levels 1–3 of the bundled Hendrycks MATH pool, spanning Algebra, Prealgebra, Intermediate Algebra, Precalculus, Number Theory, Counting and Probability, and Geometry. Levels 4–5 are not included. For `Qwen2.5-1.5B`, GRPO mode `zv_frac` is concentrated near 0.18 to 0.22, and the two consecutive upper threshold crossings required to enter CSP occur infrequently. The primary A-CSP runs therefore spend approximately 10% of training steps in CSP.

The relationship changes with model scale. On `Qwen2.5-3B` MATH-L1-3, mean `zv_frac` and CSP exposure are substantially higher, yet GRPO and A-CSP have the same mean Final EM in Table 2. This provides another example of why the controller statistic should be treated as an operational training signal rather than a sufficient predictor of benefit.

M Pair Selection Reference Runs

The pair selection reference runs differ from primary A-CSP in the pairing rule, reward representation, and entropy configuration. They are therefore not controlled pair selection ablations and are included only for completeness. We draw no causal conclusion from their numerical differences.

Table M.11: **Pair selection reference runs on Qwen2.5-1.5B GSM8K** (H100, three seeds). These runs are not configuration matched and are reported only as reference runs.

Pair selection	Peak EM	Final EM	Ret.
Reward extremes (primary η=0.001)	0.730 $\pm$0.020	0.707 $\pm$0.029	0.97
Random (IRPO style, η=0.01)	0.743 $\pm$0.006	0.697 $\pm$0.025	0.94

N Full Hyperparameter Table

Table N.12: **Training and evaluation hyperparameters** (H100 rerun).

Setting	Value
Hardware	One H100 GPU allocated per Ray worker (AWS `p5.48xlarge`)
Models	`Qwen2.5-1.5B/3B-Instruct`, transfer: SmolLM2-1.7B, Llama-3.2-3B
Datasets	GSM8K and MATH-L1-3 custom pool, 500 training and 100 evaluation prompts per seed (MATH-L1-3 is not canonical MATH500)
Adapter	LoRA r=16, α=32 on q, k, v, o projections
Optimizer	AdamW, lr 3×10^{-5}, weight decay 0.01
Horizon	1,000 steps, eval at step 1 and every 50 steps (n_{val}=100)
Rollouts	G=8 completions per prompt, batch 8 prompts
GRPO	clip ϵ=0.2, KL coef. 0.04 toward SFT initialization
CSP	β=0.1, SFT weight α=0.5, entropy coef. η=0.001 (A-CSP primary) or η=0.01 (always on CSP)
Switching	θ_{up}=0.30, θ_{down}=0.20, 2 consecutive high evals to enter CSP
Rewards	Shaped verifier per Table 1, EM eval uses exact match
Seeds	$\{42, 123, 7\}$ unless noted