TANDEM REINFORCEMENT LEARNING WITH VERIFIABLE REWARDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) has driven substantial gains in language-model reasoning. However, models fine-tuned with RLVR need not remain compatible with weaker partners: gains achieved alone may erode under collaboration, while policy drift may place reasoning beyond a weaker model's predictive reach. To address both problems, we introduce Tandem RLVR (TRLVR), bringing the recently proposed tandem-training paradigm beyond proof-of-concept settings and into modern RLVR. In TRLVR, a *senior* learns from rewarded trajectories co-generated with a frozen *junior*, while the standard GRPO objective is applied only to senior-emitted tokens. Training Qwen3-4B-Instruct on competition math, TRLVR performs on par with a matched GRPO senior when reasoning alone. We characterize compatibility through the complementary lenses of handoff robustness and junior legibility. Under reasoning-step handoffs, the TRLVR senior retains nearly all of its solo performance with the junior, reducing the *communication tax* to a negligible loss. The senior's reasoning is also more legible to the junior: its token distribution stays more closely anchored to the junior's, and the junior predicts the senior's chain-of-thought more readily token by token. An ablation that explicitly regularizes GRPO toward the junior does not recover the same combination of capability and compatibility. These results provide initial evidence that tandem rollout structure is a promising direction for keeping RLVR reasoning gains accessible to weaker partners.[1]

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has emerged as a dominant post-training paradigm in language-model post-training. In RLVR, a model samples a solution, an external verifier scores the final answer, and policy optimization reinforces successful trajectories. It effectively produces large gains in reasoning performance across various challenging domains (Shao et al., 2024; Guo et al., 2025; Yu et al., 2025).

However, this objective rewards correct answers without requiring reasoning that weaker agents can follow. As a model becomes better at solving problems, its reasoning can become harder for its pre-RL base model, weaker partners, or human overseers to predict, continue, or understand. Models that excel alone can fail to retain that performance when collaborating with weaker partners (Davidson et al., 2025). RLVR also shifts the policy away from its base distribution and, in some reported cases, produces idiosyncratic token patterns rather than transparent explanations (Guo et al., 2025; Kirchner et al., 2024; Karpov et al., 2025; Skaf et al., 2025).

This incompatibility need not reflect a defect in capability, since token-level analyses attribute part of RLVR's reasoning gains to precisely the positions where the trained policy departs furthest from the base (Li et al., 2025a; Meng et al., 2026). This suggests that the model does not reason well in spite of its private dialect but partly because of it; the illegibility is tied to the capability itself (Jose, 2025). GPT-6 Astra, for example, solves ARC-AGI-3 games while planning in an algebraic shorthand of its own invention (ARC Prize Foundation, 2026). Reasoning traces extracted from deployed frontier models also reveal stretches of obfuscated reasoning in challenging tasks (Panfilov et al., 2026). OpenAI describes this widening gap between capability and human understanding as an "alien mind," arguing that model behavior becomes harder to interpret as capability grows (Pachocki, 2026).

[1]The codebase is available at `https://anonymous.4open.science/r/Tandem-RLVR-86BB/`.

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.

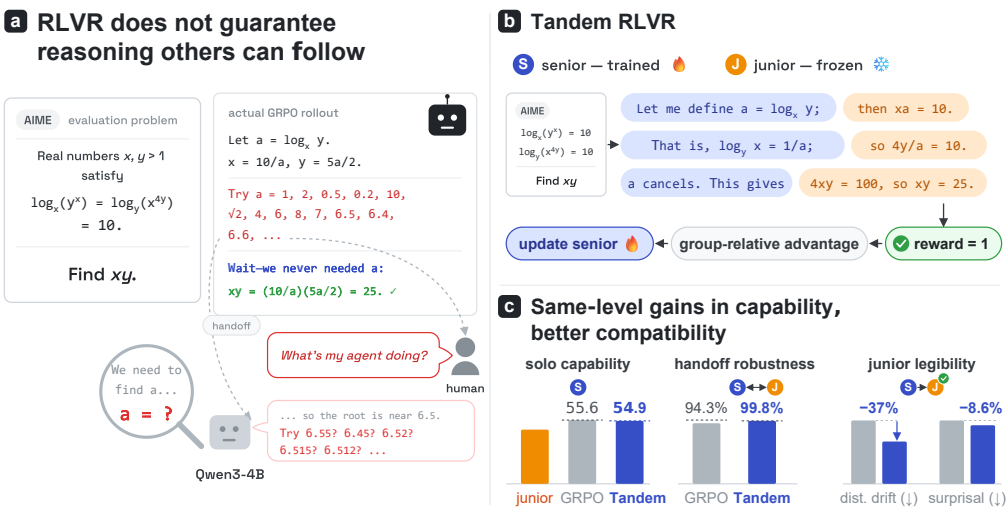

Figure 1: (a) In an AIME example from our evaluation, a GRPO-trained model repeatedly guesses an unnecessary intermediate value before realizing that it cancels, producing a chain of thought that weaker partners and overseers find hard to follow. (b) A glance at our methodology, Tandem RLVR (TRLVR). (c) On competition math, the Tandem RLVR senior matches GRPO in solo reasoning capability, retains nearly all of that capability when the junior takes over half the reasoning, and expresses its reasoning in language the junior can follow.

There are thus two reasons for concern. We build AI for collaboration with people and weaker systems (National Academies of Sciences, Engineering, and Medicine, 2022), but a driving policy, for example, can outperform the driver it assists and still be unsafe when that driver cannot anticipate it (Teoh & Kidd, 2017). We also want to learn from AI beyond merely delegating work to it (Shin et al., 2023). Knowledge transfer and oversight depend on being able to make sense of a model's reasoning. A chain of thought that no one can parse offers nothing to learn from, and it can also make weak-to-strong supervision (Burns et al., 2023) more difficult.

We ask whether RLVR can improve capability without eroding compatibility. Tandem training, a recently introduced paradigm, offers a way to train for compatibility: a trainable *senior* model co-generates each rollout with a frozen *junior* partner and learns from their joint reward (West et al., 2026). Proofs of concept exist in chess (Hamade et al., 2024) and a simplified LLM reasoning setting (West et al., 2026), but it remains unclear whether tandem training can preserve the capability gains of modern RLVR while making those gains accessible to weaker partners. In this work, we adapt tandem training to RLVR as Tandem RLVR (TRLVR). The senior and junior start from the same base model and co-generate each response by stochastically alternating at word boundaries (Figure 1b). A binary verifier then scores the completed response, and the standard GRPO objective updates the senior on its own tokens. The reward, verifier, and policy-gradient loss are unchanged from the GRPO baseline, isolating rollout structure as the training difference.

With Qwen3-4B-Instruct on competition math, TRLVR preserves the capability gains of RLVR: its senior performs on par with a matched GRPO senior when reasoning alone. We then assess compatibility along two complementary dimensions: *handoff robustness* and *junior legibility*. Handoff robustness measures how much of the senior's solo performance is retained when a weaker partner joins the reasoning process. Under reasoning-step handoffs, the TRLVR senior retains 99.8% of its solo pass@4 on macro average, compared with 94.3% for its GRPO counterpart, reducing the *communication tax* to a negligible loss (Davidson et al., 2025). Junior legibility measures how well the junior can predict the senior's reasoning. Compared with GRPO, the TRLVR senior stays closer to the junior's token distribution, and its reasoning traces are more predictable to the junior token by token. An ablation that explicitly regularizes GRPO toward the junior does not recover the same combination of capability and compatibility.

Taken together, these findings suggest that tandem rollouts are a promising direction for keeping RLVR gains accessible to weaker partners. We view TRLVR as an initial step toward training for

both handoff robustness and junior legibility alongside capability, with a broad design space left to explore.

2 Related work

Tandem training. Hamade et al. (2024) introduced tandem training in chess, showing that training a model to win alongside a weaker partner differs from training it to win alone. West et al. (2026) extended tandem training to language modeling. On GSM8K, randomized handoffs to a frozen junior during RL taught the senior to reduce jargon and adapt its language while preserving task accuracy. These studies did not evaluate tandem training with the long chains of thought and policy optimization involved in modern RLVR. We study tandem training in this modern RLVR setting with TRLVR and evaluate both solo capability and compatibility, using handoff robustness and junior legibility to measure the compatibility.

Compatibility concerns in RLVR. RLVR rewards verified task success without requiring either that the learned capability remain usable when a weaker partner joins the reasoning process or that the reasoning trace remain within that partner's predictive reach. More broadly, models that perform well alone can degrade sharply in heterogeneous collaboration, a phenomenon termed the *collaboration gap* (Davidson et al., 2025; Wynn et al., 2025). Within RLVR, DeepSeek-R1-Zero exhibits reduced readability and unprompted language mixing (Guo et al., 2025), and RLVR can induce Chinese–English code-switching in bilingual reasoners (Li et al., 2025a); related work further identifies steganographic reasoning patterns inaccessible to downstream monitors (Karpov et al., 2025; Skaf et al., 2025; Baker et al., 2025; Drori et al., 2026). Recent evaluations of weak-to-strong legibility likewise find that strong reasoning performance need not produce traces useful to weaker models (Roytburg et al., 2026; Jose, 2025; Li et al., 2025b). Yet not all distribution shift is pathological: token-level interventions indicate that a small set of high-divergence positions can carry RLVR's capability gains (Meng et al., 2026). The target is therefore not to suppress drift indiscriminately, but to preserve capability while keeping it accessible to weaker partners.

Approaches to compatibility and legibility. When the desired behavior can be specified in advance, textual principles can prescribe it (Bai et al., 2022), supervised reasoning traces can instantiate a target solution style (Magister et al., 2023; Li et al., 2025b; Cetin et al., 2025), and process supervision can reward valid intermediate steps (Uesato et al., 2022; Lightman et al., 2023; Anwar et al., 2026; Surana et al., 2026a). Prover–verifier games train for solution checkability by a learned verifier (Kirchner et al., 2024). These approaches are complementary to tandem training. Specification-based methods are well suited when a target behavior is available, while prover–verifier methods operationalize legibility as checkability by a designated verifier. Tandem training uses a different interaction-based criterion: whether a designated weaker generator can take over the senior's partial trajectory and still complete the task successfully. Because the task verifier rewards the co-generated trajectory as a whole, tandem training requires neither a canonical surface form nor additional process labels.

3 Tandem RLVR

3.1 Preliminaries: tandem training

Tandem training (Hamade et al., 2024; West et al., 2026) is a recently introduced paradigm for training models that remain effective when paired with a partner. It uses task performance on jointly generated rollouts as the training signal, so compatible reasoning need not be specified in advance. By comparison, explicit methods such as system prompts or supervised finetuning on canonical solutions require an *a priori* specification that depends on deployment context (West et al., 2026; Bai et al., 2022; Lightman et al., 2023).

A trainable *senior* policy π_{sen} and a frozen *junior* policy π_{jun} stochastically alternate generations to produce each rollout. At predetermined handoff boundaries such as tokens, words, or sentences, a coin flip determines the next active model. For example, token-level handoffs use a new coin flip for each token. A single reward function scores the completed rollout as one trajectory, and a policy-gradient algorithm updates the senior using that reward while the junior remains unchanged. However, success

under these handoffs does not by itself establish that the junior semantically understands the senior's reasoning. Task success may arise from information uptake, division of labor, error correction, or partially independent reasoning.

In TRLVR, we initialize the senior and junior from the same base model (§3.2) and co-generate rollouts with word-level handoffs (§3.3). The senior is then updated with Group Relative Policy Optimization (GRPO) (Shao et al., 2024), with the loss applied only to senior-emitted tokens (§3.4).

3.2 TANDEM PAIRS

We use the senior's own pre-RL initialization as the junior to study whether RLVR can improve capability while preserving compatibility with the original policy. The two models begin as identical copies, but the senior becomes stronger through RLVR while the junior remains frozen. The capability gap therefore develops during training. This lets us examine how RLVR changes compatibility without introducing an initial difference in capabilities or reasoning style between the models. Because each reward is computed on a jointly generated response, the senior's updates continue to depend on how well it works with the frozen junior as its capabilities improve. Our goal is for the senior to improve its capabilities while minimizing drift from its initial state, so that it remains compatible and intelligible to the junior. We therefore initialize π_{sen} and π_{jun} from the same base model and freeze π_{jun} for the duration of training.

3.3 TANDEM ROLLOUT

A tandem rollout produces a shared response $y_{1:T}$ from prompt x, where each step t is generated by an active model $a_t \in \{\text{sen}, \text{jun}\}$. At each step, both models receive the context $x \cdot y_{<t}$, the active model samples $y_t \sim \pi_{a_t}(\cdot \mid x \cdot y_{<t})$, and the chosen y_t is appended and then fed to both models so that both states condition on the same history. Following West et al. (2026), the active model is redrawn by an independent fair coin flip at every word boundary: if y_t begins a new orthographic word, a_{t+1} is set to the senior with probability p and to the junior otherwise; if y_t does not begin a new word, $a_{t+1} = a_t$. We use $p = 0.5$ in consistency with original tandem training. Details of this setup are included in Appendix B.1, alongside a summary of the method in Figure 5.

We choose to stochastically alternate at word-level granularity over alternative choices (e.g., token-level or sentence-level). First, at sentence- or paragraph-level the senior can dominate a rollout simply by suppressing boundary tokens (e.g., end-of-sentence or end-of-line tokens) that would force a potential handoff[2], collapsing the schedule back to GRPO. Second, at token level, alternation at subword granularity disrupts coherent utterances before they form. In contrast, word-level boundaries fire on every natural-language word and are difficult to suppress without sacrificing fluency, making them the coarsest granularity the senior cannot trivially erase by output choice.

3.4 TANDEM POLICY UPDATE

Each tandem rollout receives a single binary reward $r(y) \in \{0, 1\}$ from a math verifier, computed on the full response $y_{1:T}$ regardless of which model emitted each token. The senior is updated with the standard GRPO objective (Shao et al., 2024), with the response mask set to select only senior-emitted positions; the per-token policy-gradient contribution is

$$\mathcal{L}_{\text{TRLVR}}(sen) \;=\; -\,\mathbb{E}\left[\sum_{t\,:\,a_t=\text{sen}} A_t \,\log \pi_{\text{sen}}(y_t \mid x \cdot y_{<t})\right], \tag{1}$$

where A_t is the standard GRPO group-relative advantage; the clipped surrogate ratio and KL term are also identical to GRPO and omitted here for brevity. Junior-emitted tokens enter the response, and so affect the reward the senior receives, but contribute no gradient.

Specifically, while the original tandem-training formulation also includes a soft junior-imitation term that pulls π_{sen} toward π_{jun} on junior-emitted positions with coefficient λ_{jun}, we set $\lambda_{\text{jun}} = 0$.

[2]This reward-hacking failure mode is induced by the schedule itself: as the senior strengthens, suppressing boundary tokens becomes a low-cost route to extending its own control and avoiding handoffs.

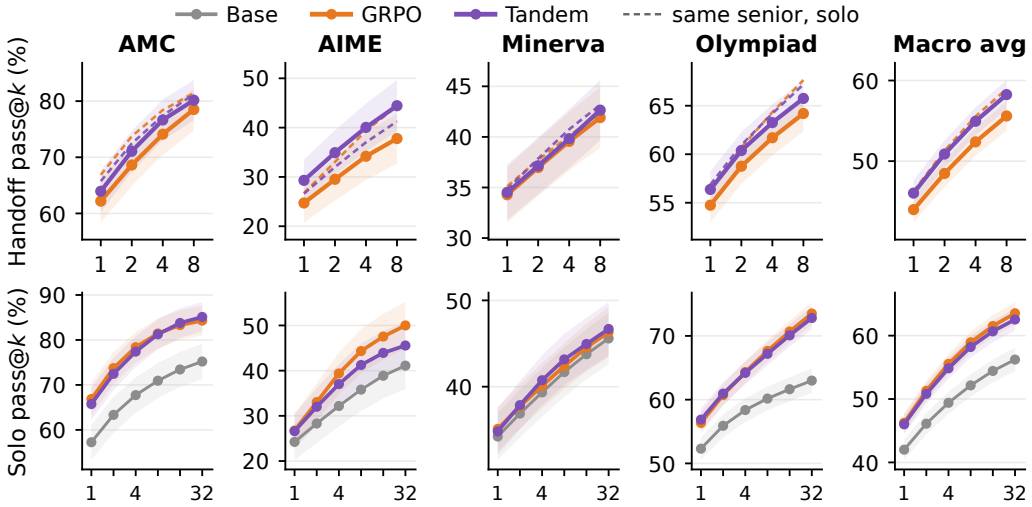

Figure 2: Reasoning-step handoff pass@k (top) and solo pass@k (bottom) of Qwen3-4B-Instruct and its GRPO- and TRLVR-trained seniors on competition math benchmarks (↑).

Beyond removing a hyperparameter we have no principled value for, this keeps the per-token loss formally the same as GRPO. Thus, the only difference between the TRLVR and GRPO objectives is the rollout structure that produces y, not an auxiliary regularization term. We evaluate each run at its best validation checkpoint under its own rollout protocol, and also compare the two at a matched training step.

The end-to-end TRLVR pipeline is illustrated in Algorithm 1 in Appendix B.4.

4 Results

4.1 Experimental setup

We fine-tune Qwen3-4B-Instruct-2507 (Yang et al., 2025) on DeepScaleR (Tan et al., 2025) with a binary correctness reward on the boxed final answer. Our baseline is GRPO (Shao et al., 2024) on the same base model, with optimization settings shared with TRLVR (Appendix B.3). We evaluate on AMC 23–25, AIME 24–26, Minerva Math (Lewkowycz et al., 2022), and Olympiad-Bench (He et al., 2024), and report pass@k via an unbiased estimator following Chen et al. (2021). We estimate bootstrap standard errors by resampling evaluation problems, reported as $\pm$ values in tables and as shaded bands in figures.

This section is organized in three parts. In §4.2, we measure the resulting models' solo reasoning capabilities, examining whether the tandem rollout preserves the reasoning gains delivered by GRPO. We then turn to handoff robustness in §4.3, the central behavioural property tandem training was originally proposed to elicit; we pair the trained models with the junior model at inference time and evaluate the team's performance. Finally, in §4.4 we ask whether the senior's chain-of-thought stays legible to the junior, measuring both distributional drift from the base and how predictable the junior finds the chain-of-thought token by token.

4.2 TRLVR retains the reasoning gains of GRPO

We first measure each model's solo reasoning capability, with no junior in the loop at test time, to test whether the tandem rollout structure preserves the gains of standard RLVR training. Each senior generates n independent samples per problem; we report pass@k via the unbiased estimator for $k \in \{1, 2, 4, 8, 16, 32\}$.

The bottom row of Figure 2 shows pass@k curves for the pre-RL base and the seniors trained with GRPO and TRLVR across benchmarks and their macro average. Two observations stand

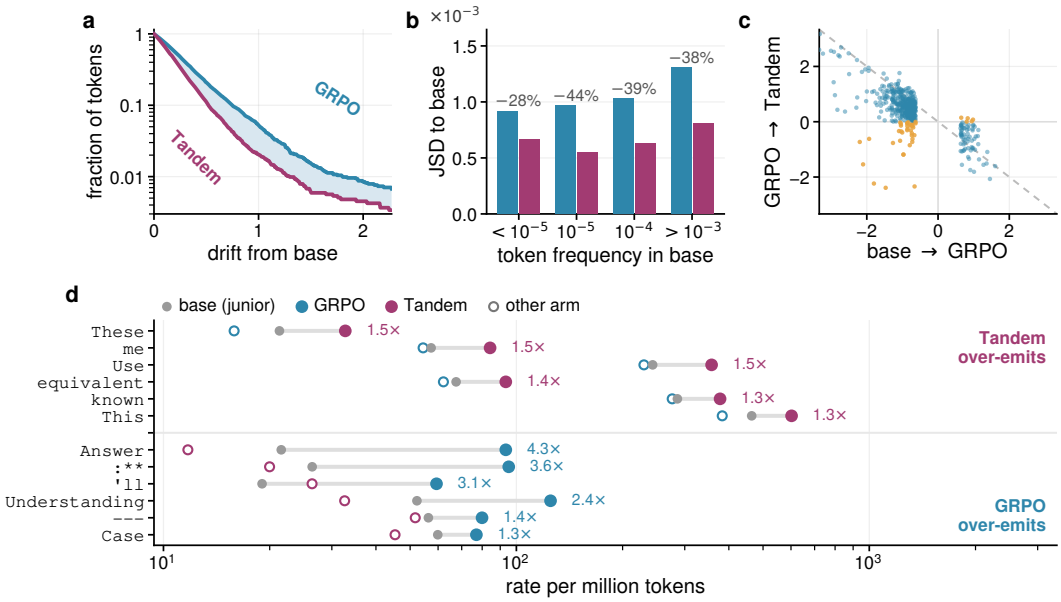

Figure 3: Distributional deviation of the GRPO- and TRLVR-trained seniors from the junior: (a) survival curve of the absolute per-token log-ratio, (b) Jensen–Shannon divergence to the junior by token frequency in the junior, (c) the GRPO senior's token-wise displacement from the junior against the TRLVR senior's displacement from GRPO, for the tokens the GRPO senior displaces most, and (d) emission rate of the tokens each senior over-emits most.

out. First, both trained seniors outperform the base model across the full k range on all four benchmarks, confirming that RLVR delivers a non-trivial capability gain. Second, the GRPO and TRLVR curves track each other closely throughout, with overlapping bootstrap uncertainty bands and no consistent direction across benchmarks. Together, these results provide empirical evidence that TRLVR preserves the *solo* reasoning gains achieved by GRPO.

4.3 TRLVR OFFERS BETTER HANDOFF ROBUSTNESS

In §3.3 we argued against training with sentence- or paragraph-level handoffs, on the grounds that the senior could game such schedules by suppressing the boundary tokens that would force a switch. For evaluation, however, there is no risk of this type of gaming behavior, and we want to measure how cooperation between language models actually unfolds in practice. In multi-agent reasoning systems, such as drafter-target setups for inference and human-AI handoffs, the natural unit of communication is a reasoning step rather than an individual word, with one party producing a step and a partner picking up where it left off (Chu et al., 2025b; Wang et al., 2025; Davidson et al., 2025)[3]. This motivates our evaluation of handoff robustness under a reasoning-step schedule: at inference, the senior and the junior alternate at every `\n\n`-boundary token, each producing one reasoning step before yielding control to the other, with the team-produced response scored as a whole.

The top row of Figure 2 shows handoff pass@k against each senior's own solo pass@k, drawn dashed, so the gap between a senior's two curves is what handing off costs it. TRLVR leads on all four benchmarks and at every k. The margin is widest on AIME, where it reaches $+6.6$ points at $k{=}8$, and on macro average it widens with k throughout. We call the team's pass@k *handoff accuracy*, and measure alongside it *handoff robustness*: handoff accuracy as a percentage of the same senior's solo pass@k, computed from the macro-averaged accuracies. The ratio measures retention of solo performance under handoffs, though it is read alongside solo capability, since a senior that scores lower alone has less to lose. At $k{=}4$ the GRPO senior retains 94.3% of its solo score and the TRLVR senior 99.8%. Together with the solo results in §4.2, these findings show that the TRLVR senior is no less capable alone than the GRPO senior and barely loses any of that capability when the junior takes

[3] also resembling a *ply* of chess games in Hamade et al. (2024).

Table 1: Junior legibility of senior generated chain-of-thoughts (Qwen3-4B-Instruct junior). **Bold** marks the better model per cell.

Metric	Model	AMC	AIME	Minerva	Olympiad	Avg.
CE (nats, ↓)	GRPO	$0.119_{\pm 0.004}$	$\mathbf{0.147}_{\pm 0.006}$	$0.113_{\pm 0.003}$	$0.139_{\pm 0.003}$	$0.129_{\pm 0.002}$
	TRLVR	$\mathbf{0.107}_{\pm 0.005}$	$\mathbf{0.147}_{\pm 0.007}$	$\mathbf{0.093}_{\pm 0.003}$	$\mathbf{0.126}_{\pm 0.003}$	$\mathbf{0.118}_{\pm 0.002}$
	Δ	$\mathbf{-0.012}_{\pm 0.006}$	$+0.000_{\pm 0.009}$	$\mathbf{-0.020}_{\pm 0.004}$	$\mathbf{-0.013}_{\pm 0.004}$	$\mathbf{-0.011}_{\pm 0.003}$
α (%, ↑)	GRPO	$95.89_{\pm 0.12}$	$95.38_{\pm 0.15}$	$96.08_{\pm 0.10}$	$95.40_{\pm 0.07}$	$95.69_{\pm 0.06}$
	TRLVR	$\mathbf{97.29}_{\pm 0.07}$	$\mathbf{96.87}_{\pm 0.09}$	$\mathbf{97.53}_{\pm 0.05}$	$\mathbf{97.06}_{\pm 0.04}$	$\mathbf{97.19}_{\pm 0.03}$
	Δ	$\mathbf{+1.40}_{\pm 0.14}$	$\mathbf{+1.48}_{\pm 0.17}$	$\mathbf{+1.45}_{\pm 0.11}$	$\mathbf{+1.66}_{\pm 0.08}$	$\mathbf{+1.50}_{\pm 0.06}$

over half of the reasoning. We report paired t-tests on AMC, AIME, and Minerva for this gap, and for the further TRLVR gains reported in the following subsection, in Appendix C.1.

4.4 TRLVR KEEPS THE SENIOR'S CHAIN-OF-THOUGHT LEGIBLE TO THE JUNIOR

We now ask whether the senior's chain-of-thought stays legible to the junior, in two senses. The first is distributional, whether the senior's overall token usage stays near the junior's or drifts away as documented for RLVR at scale (§2). The second is sequential, whether the junior, reading the senior's chain-of-thought left to right, finds each next token predictable. The two measures capture different aspects of legibility, since similar marginal token distributions do not necessarily make the senior's next token predictable to the junior. During tandem rollouts, the frozen junior can take over at word boundaries, so its continuations affect the reward used to update the senior. We therefore expect training to favor reasoning that stays closer to the junior's distribution and is more predictable to it.

4.4.1 TRLVR CURBS DISTRIBUTIONAL DRIFT

We estimate each senior's marginal token distribution from its generations on the evaluation sets, and measure per-token deviation from the junior by the log-ratio $\log(p_{\text{sen}}(t) \,/\, p_{\text{jun}}(t))$; Jensen–Shannon divergence (JSD) to the junior aggregates these deviations, weighted by how often each token is actually used. Panels (a) and (b) of Figure 3 show that the TRLVR senior stays closer to the junior's token distribution than the GRPO senior. Its survival curve lies below GRPO's at every deviation threshold, its JSD to the junior is 37% lower, and the gap persists in every band when the divergence is decomposed by token frequency in the junior, so the reduction comes neither from a few common tokens nor from the long tail. Panel (c) makes the comparison token by token: among the top tokens GRPO displaces most, 91% move back toward the junior under TRLVR, and the two moves are anti-correlated (Spearman $\rho = -0.60$), so the further GRPO displaces a token, the more of that displacement TRLVR tends to undo.

Panel (d) provides a case study of the most displaced tokens under each recipe. From the tokens each senior over-emits we keep the interpretable ones, ranked by how strongly they are over-emitted. GRPO's extra mass goes to the markup of a formatted report, headings, bold labels, and horizontal rules; the TRLVR senior's extra mass goes to connective phrases that carry one step into the next. Both methods reward the final answer, but in TRLVR the junior receives the senior's intermediate reasoning as context for its own continuation. This may help explain why TRLVR more often produces phrases that connect successive reasoning steps. Appendix C.2 extends the case study to the phrase level.

4.4.2 TRLVR REDUCES THE JUNIOR'S SURPRISAL

We measure the sequential sense with two analyses, summarized in Table 1. The first is the junior's per-token cross-entropy on the senior's chain-of-thought: the average *nats* required to encode each senior token under the junior's predictive distribution, where lower means the junior is less surprised. The TRLVR senior's cross-entropy is lower by 0.011 nats on average, with clear gains on AMC, Minerva, and OlympiadBench and a match on AIME. This corresponds to an 8.6% reduction in per-token surprisal on average, reaching 17% on Minerva.

Appendix C.4 provides two controls on this metric. Lower junior cross-entropy is associated with correctness within the same problem, and the TRLVR advantage grows after subtracting each trace's

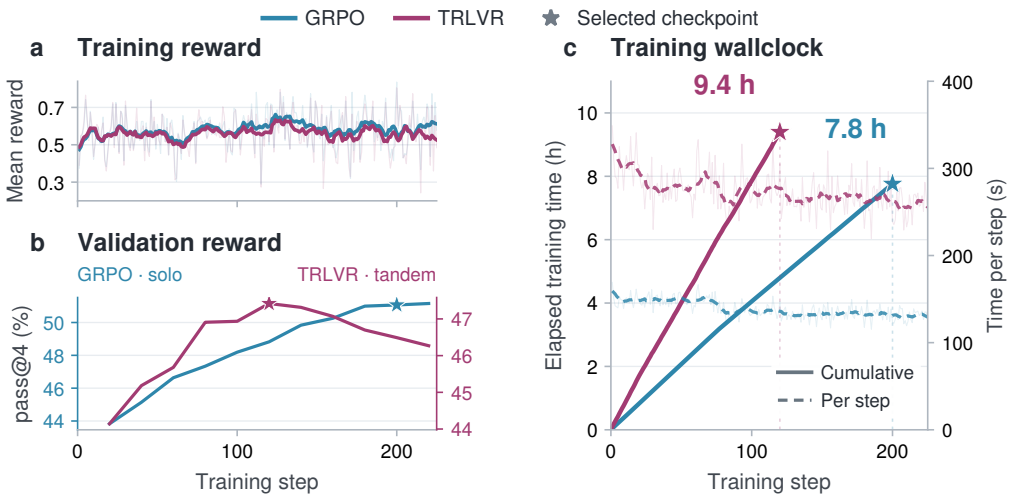

Figure 4: Training dynamics of TRLVR and GRPO. (a) Mean reward. (b) Validation pass@4 under each run's evaluation protocol. (c) Accumulated wallclock to selected checkpoints (solid, left axis) and per-step wallclock (dashed, right axis).

cross-entropy under its generating senior, ruling out greater predictability under the senior as the explanation for the gap.

The second is the token-level distribution overlap $\alpha = \sum_v \min(p_{\text{sen}}(v \mid \cdot), p_{\text{jun}}(v \mid \cdot))$ averaged over all positions (Leviathan et al., 2023): the probability mass both models assign to the same tokens at each step, ranging from 0 (fully disjoint) to 1 (identical), and reported as a percentage. As shown, the TRLVR senior's distribution overlap is uniformly higher across all four benchmarks. On average, the overlap increases by 1.50 percentage points, from 95.69% for GRPO to 97.19% for TRLVR. The remaining gap to full overlap falls from 4.31% to 2.81%, a reduction of roughly 35%.

Section takeaway. The TRLVR senior retains the solo reasoning gains of GRPO (§4.2), preserves nearly all of that performance when the junior takes over half the reasoning (§4.3), and keeps its reasoning in language the junior can follow (§4.4).

5 Discussion

5.1 Training dynamics of TRLVR

TRLVR generates each rollout with two models, which raises questions about training stability and computational cost. Figure 4 reports training reward, validation performance, and wallclock time. For these runs, we select TRLVR checkpoints by tandem-rollout validation pass@4 and GRPO checkpoints by solo validation pass@4. This keeps TRLVR checkpoint selection consistent with its cooperative training objective. Panel (b) therefore uses separate axes for the two validation curves, which measure performance under different rollout protocols.

Panel (a) shows that training reward follows a stable trajectory under TRLVR, closely tracking GRPO throughout. These reward curves suggest that tandem rollouts preserve stable learning in this setting. Running both models at each generation step does add computational cost. Our initial implementation could not support the long contexts required for RLVR, so we extended the vLLM backend (Kwon, 2025) to run tandem generation within its inference engine (Appendix B.1). Each model maintains its own paged KV cache for the shared response, avoiding prefix recomputation at handoffs. The selected TRLVR checkpoint is at step 120, compared with step 200 for GRPO, partially offsetting the per-step overhead. Training time to these checkpoints is 9.4 hours for TRLVR versus 7.8 hours for GRPO (panel c).

We also evaluate GRPO at step 120, the step of the selected TRLVR checkpoint (matched step in Table 2). At the matched step, TRLVR scores higher alone and under handoff, with handoff

Table 2: GRPO at its best validation step and at TRLVR's matched step, KL-regularized GRPO at two coefficients, and TRLVR on the evaluation axes of §4 (macro averaged).

Method	Solo Capability (pass@4, %, ↑)	Handoff Accuracy (pass@4, %, ↑)	Handoff Robustness (%, ↑)	Junior Legibility (nats, ↓)
GRPO	$55.56_{\pm 2.1}$	$\underline{52.40}_{\pm 2.3}$	94.3	$0.129_{\pm 0.002}$
GRPO (matched step)	$53.14_{\pm 1.7}$	$50.60_{\pm 1.8}$	95.2	$\underline{0.114}_{\pm 0.002}$
KL-Reg, β=10^{-3}	$55.24_{\pm 2.2}$	$52.09_{\pm 2.3}$	94.3	$0.131_{\pm 0.003}$
KL-Reg, β=10^{-1}	$50.15_{\pm 2.3}$	$49.15_{\pm 2.3}$	$\underline{98.0}$	$\mathbf{0.114}_{\pm 0.003}$
TRLVR	$54.85_{\pm 2.1}$	$\mathbf{54.74}_{\pm 2.4}$	**99.8**	$0.118_{\pm 0.002}$

robustness of 99.8% versus 95.2%. Junior cross-entropy is slightly lower for GRPO there, as is capability: within the GRPO run, shorter training trades capability for legibility. TRLVR improves on that trade-off, being more capable than GRPO at the matched step, more legible than GRPO at its best validation step, and more robust under handoff than either.

5.2 Ablation study: KL-regularization towards the junior

We next test whether policy-level KL regularization toward the junior achieves the anchoring that TRLVR obtains through its rollout structure. TRLVR anchors the senior only implicitly, through co-generation under a shared reward, so an explicit per-token KL penalty toward the same junior is the natural ablation. In our setup this baseline is exactly standard KL-regularized GRPO: the recipe regularizes toward the pre-RL reference policy, which is the same frozen checkpoint that serves as TRLVR's junior. We denote this baseline *KL-Reg* and train it at the documented default coefficient of $\beta = 10^{-3}$ and at $\beta = 10^{-1}$, using the same training data, optimizer settings, and evaluation protocol as TRLVR and GRPO.

Table 2 reports macro averages for solo capability, handoff accuracy, and junior legibility, together with handoff robustness. At the default strength, KL-Reg preserves solo capability but leaves handoff accuracy and robustness at the GRPO level, without improving junior legibility. At $\beta = 10^{-1}$, KL-Reg has the lowest junior cross-entropy in the table, but solo capability drops by more than five percentage points relative to GRPO and handoff accuracy is also the lowest. Its 98.0% handoff robustness indicates that it retains most of this reduced solo capability. In comparison, TRLVR retains 99.8% under handoff while preserving GRPO-level solo capability.

These results are consistent with the difference in how the two objectives use the junior. KL-Reg measures next-token divergence from the junior on the senior's own rollouts and receives task rewards from senior-only responses. In TRLVR, junior continuations form part of the rewarded trajectory, so the senior's updates depend on how the pair completes the task. Appendix D details this difference between the training signals.

6 Conclusion

RLVR rewards solving problems alone and leaves open whether the gains survive a weaker partner joining the reasoning. We introduced TRLVR, which co-generates each rewarded trajectory with a frozen copy of the pre-RL base and applies the standard GRPO objective to senior tokens only, leaving reward, verifier, and loss unchanged. Across four competition-math benchmarks, the senior matches GRPO alone, keeps 99.8% of its macro-average solo pass@4 under reasoning-step handoffs against 94.3% for GRPO, and stays more legible to the junior, in token distribution and in next-token predictability.

Several directions for future work are worth highlighting. First, replacing the fixed self-paired junior with a pool of diverse juniors varying in capability, style, language, or tool use could regularize the senior toward more broadly intelligible behavior. Second, pairing the senior with juniors calibrated to specific human skill levels would provide a more direct bridge to the genuine human-AI handoff scenarios that originally motivated the tandem framework. We view this direction as currently bottlenecked by the absence of comparably well-calibrated, human-skill-level proxies for general-purpose language models, and a promising avenue once such models mature.

ETHICS STATEMENT

This work studies whether reinforcement learning with verifiable rewards can improve model's reasoning capability while preserving compatibility with weaker model partners. Our experiments use publicly available models and datasets and do not involve human subjects or private data. We do not identify additional dual-use concerns specific to this study beyond those inherent to general-purpose language model research.

REPRODUCIBILITY STATEMENT

Appendix B.1 describes the tandem rollout implementation, and Appendix B.4 provides the training pseudocode. Dataset sources and preparation are described in Appendix B.2, and training hyperparameters and hardware are listed in Appendix B.3 and Table 3. Section 4 defines the evaluation protocols and metrics. Code for data preparation, training, and evaluation is available at `https://anonymous.4open.science/r/Tandem-RLVR-86BB/`.

AI USE STATEMENT

Generative AI assisted with writing, reference checks, figures, and formatting. The authors are responsible for the final content.

REFERENCES

Usman Anwar, Tim Bakker, Dana Kianfar, Cristina Pinneri, and Christos Louizos. Analyzing and improving chain-of-thought monitorability through information theory. *arXiv preprint arXiv:2602.18297*, 2026. URL https://arxiv.org/abs/2602.18297.

ARC Prize Foundation. OpenAI's GPT-6 Astra on ARC-AGI-3. https://arcprize.org/blog/astra, 2026.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y. Guan, Aleksander Madry, Wojciech Zaremba, Jakub Pachocki, and David Farhi. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation. *arXiv preprint arXiv:2503.11926*, 2025. URL https://arxiv.org/abs/2503.11926.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*, 2023.

Edoardo Cetin, Tianyu Zhao, and Yujin Tang. Reinforcement learning teachers of test time scaling. *arXiv preprint arXiv:2506.08388*, 2025. URL https://arxiv.org/abs/2506.08388.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. Gpg: A simple and strong reinforcement learning baseline for model reasoning. *arXiv preprint arXiv:2504.02546*, 2025a.

Yuanlin Chu, Bo Wang, Xiang Liu, Hong Chen, Aiwei Liu, and Xuming Hu. Ssr: Speculative parallel scaling reasoning in test-time. *arXiv preprint arXiv:2505.15340*, 2025b.

Tim R Davidson, Adam Fourney, Saleema Amershi, Robert West, Eric Horvitz, and Ece Kamar. The collaboration gap. *arXiv preprint arXiv:2511.02687*, 2025.

Jacob Drori, Luke Marks, Bryce Woodworth, Alex Cloud, and Alexander Turner. Output supervision can obfuscate the chain of thought. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/53813b70e97b6e5778e24fd88e1259f1-Abstract-Conference.html.

Tianyu Fu, Zihan Min, Hanling Zhang, Jichao Yan, Guohao Dai, Wanli Ouyang, and Yu Wang. Cache-to-cache: Direct semantic communication between large language models. *arXiv preprint arXiv:2510.03215*, 2025.

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

Karim Hamade, Reid McIlroy-Young, Siddhartha Sen, Jon Kleinberg, and Ashton Anderson. Designing skill-compatible ai: Methodologies and frameworks in chess. *arXiv preprint arXiv:2405.05066*, 2024.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

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, 2021. URL https://arxiv.org/abs/2103.03874. NeurIPS 2021 Datasets and Benchmarks Track.

Difan Jiao, Yilun Liu, Zhenwei Tang, Daniel Matter, Jürgen Pfeffer, and Ashton Anderson. SPIN: Sparsifying and integrating internal neurons in large language models for text classification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 4666–4682. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.277. URL https://aclanthology.org/2024.findings-acl.277/.

Difan Jiao, Yilun Liu, Ye Yuan, Zhenwei Tang, Linfeng Du, Haolun Wu, and Ashton Anderson. LLM safety from within: Detecting harmful content with internal representations. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 39711–39727. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.acl-long.1844. URL https://aclanthology.org/2026.acl-long.1844/.

Difan Jiao, Qianfeng Wen, Blair Yang, Zhenwei Tang, and Ashton Anderson. ThinkTwice: Jointly optimizing large language models for reasoning and self-refinement. *arXiv preprint arXiv:2604.01591*, 2026b.

Arun Jose. Reasoning models sometimes output illegible chains of thought. *arXiv preprint arXiv:2510.27338*, 2025.

Artem Karpov, Tinuade Adeleke, Seong Hah Cho, and Natalia Perez-Campanero. The steganographic potentials of language models. *arXiv preprint arXiv:2505.03439*, 2025.

Jan Hendrik Kirchner, Yining Chen, Harri Edwards, Jan Leike, Nat McAleese, and Yuri Burda. Prover-verifier games improve legibility of llm outputs. *arXiv preprint arXiv:2407.13692*, 2024.

Woosuk Kwon. *vLLM: An Efficient Inference Engine for Large Language Models*. PhD thesis, UC Berkeley, 2025.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pp. 19274–19286. PMLR, 2023.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

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

Yihao Li, Jiayi Xin, Miranda Muqing Miao, Qi Long, and Lyle Ungar. The impact of language mixing on bilingual llm reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 32519–32536, 2025a.

Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. Small models struggle to learn from strong reasoners. *arXiv preprint arXiv:2502.12143*, 2025b. URL https://arxiv.org/abs/2502.12143.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The twelfth international conference on learning representations*, 2023.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1773–1781, 2023.

Haoming Meng, Kexin Huang, Shaohang Wei, Chiyu Ma, Shuo Yang, Xue Wang, Guoyin Wang, Bolin Ding, and Jingren Zhou. Sparse but critical: A token-level analysis of distributional shifts in rlvr fine-tuning of llms. *arXiv preprint arXiv:2603.22446*, 2026.

National Academies of Sciences, Engineering, and Medicine. *Human-AI Teaming: State-of-the-Art and Research Needs*. The National Academies Press, Washington, DC, 2022. doi: 10.17226/26355.

Jakub Pachocki. An alien mind. https://openai.com/index/an-alien-mind/, 2026.

Alexander Panfilov, David Schmotz, Ilia Shumailov, Luca Beurer-Kellner, Joachim Schaeffer, Ameya Prabhu, Jonas Geiping, and Maksym Andriushchenko. Stealing reasoning traces from proprietary LLM APIs. *arXiv preprint arXiv:2608.09867*, 2026.

Dani Roytburg, Shreya Sridhar, and Daphne Ippolito. Measuring weak-to-strong legibility of reasoning models. *arXiv preprint arXiv:2603.20508*, 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.

Minkyu Shin, Jin Kim, Bas van Opheusden, and Thomas L Griffiths. Superhuman artificial intelligence can improve human decision-making by increasing novelty. *Proceedings of the National Academy of Sciences*, 120(12):e2214840120, 2023.

Joey Skaf, Luis Ibanez-Lissen, Robert McCarthy, Connor Watts, Vasil Georgiv, Hannes Whittingham, Lorena Gonzalez-Manzano, David Lindner, Cameron Tice, Edward James Young, et al. Large language models can learn and generalize steganographic chain-of-thought under process supervision. *arXiv preprint arXiv:2506.01926*, 2025.

Rohan Surana, Gagan Mundada, Xunyi Jiang, Chuhan Wang, Zhenwei Tang, Difan Jiao, Zihan Huang, Yuxin Xiong, Junda Wu, Sheldon Yu, Xintong Li, Raghav Jain, Nikki Kuang, Sizhe Zhou, Bowen Jin, Zhendong Chu, Tong Yu, Ryan Rossi, Kuan-Hao Huang, Jingbo Shang, Jiawei Han, and Julian McAuley. Generate, filter, control, replay: A comprehensive survey of rollout strategies for LLM reinforcement learning. *arXiv preprint arXiv:2605.02913*, 2026a. URL https://arxiv.org/abs/2605.02913.

Rohan Surana, Gagan Mundada, Xunyi Jiang, Chuhan Wang, Zhenwei Tang, Difan Jiao, Zihan Huang, Yuxin Xiong, Junda Wu, Sheldon Yu, et al. Generate, filter, control, replay: A comprehensive survey of rollout strategies for llm reinforcement learning. *arXiv preprint arXiv:2605.02913*, 2026b.

Sijun Tan, Michael Luo, Justin Wong, Colin Cai, Xiaoxiang Shi, William Yuan Tang, Manan Roongta, Tianjun Zhang, Li Erran Li, Raluca Ada Popa, et al. Deepscaler: Effective rl scaling of reasoning models via iterative context lengthening. 2025.

Eric R Teoh and David G Kidd. Rage against the machine? Google's self-driving cars versus human drivers. *Journal of Safety Research*, 63:57–60, 2017.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Xiao Wang, Jia Wang, Yijie Wang, Pengtao Dang, Sha Cao, and Chi Zhang. Mars: toward more efficient multi-agent collaboration for llm reasoning. *arXiv preprint arXiv:2509.20502*, 2025.

Robert West, Ashton Anderson, Ece Kamar, and Eric Horvitz. Tandem training for language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8265–8278, 2026.

Andrea Wynn, Harsh Satija, and Gillian Hadfield. Talk isn't always cheap: Understanding failure modes in multi-agent debate. *arXiv preprint arXiv:2509.05396*, 2025. URL https://arxiv.org/abs/2509.05396.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

Guanning Zeng, Zhaoyi Zhou, Daman Arora, and Andrea Zanette. Shrinking the variance: Shrinkage baselines for reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2511.03710*, 2025.

Runzhe Zhan, Yafu Li, Zhi Wang, Xiaoye Qu, Dongrui Liu, Jing Shao, Derek F Wong, and Yu Cheng. Exgrpo: Learning to reason from experience. *arXiv preprint arXiv:2510.02245*, 2025.

Yuheng Zhang, Wenlin Yao, Changlong Yu, Yao Liu, Qingyu Yin, Bing Yin, Hyokun Yun, and Lihong Li. Improving sampling efficiency in rlvr through adaptive rollout and response reuse. *arXiv preprint arXiv:2509.25808*, 2025.

Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, et al. Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*, 2025.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025a.

Haizhong Zheng, Yang Zhou, Brian R Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. Act only when it pays: Efficient reinforcement learning for llm reasoning via selective rollouts. *arXiv preprint arXiv:2506.02177*, 2025b.

Jiaru Zou, Ruizhong Qiu, Gaotang Li, Xiyuan Yang, Katherine Tieu, Pan Lu, Ke Shen, Hanghang Tong, Yejin Choi, Jingrui He, James Zou, Mengdi Wang, and Ling Yang. Latent collaboration in multi-agent systems. *arXiv preprint arXiv:2511.20639*, 2025.

A Extended Related Work

Reinforcement learning with verifiable rewards. Recent advances in reasoning post-training have leveraged reinforcement learning with verifiable rewards (RLVR). Shao et al. (2024) introduced Group Relative Policy Optimization (GRPO) as a practical recipe for this setting, and Guo et al. (2025) demonstrated that outcome-only RL can elicit strong reasoning capabilities without process supervision. A growing follow-up literature expands the GRPO design space along two axes: optimization-side variants revisit stability, optimization biases, and ratio aggregation (Yu et al., 2025; Liu et al., 2025; Zheng et al., 2025a; Zhao et al., 2025; Chu et al., 2025a; Zeng et al., 2025; Surana et al., 2026b), while rollout-side work targets sampling efficiency, prompt filtering, and reuse of high-signal trajectories (Zhang et al., 2025; Zhan et al., 2025; Zheng et al., 2025b). RLVR has also been extended beyond one-shot solution generation to jointly optimize reasoning and self-refinement under the same verifiable outcome reward (Jiao et al., 2026b). TRLVR is complementary to these directions: it changes which model emits each token within a single rollout while leaving the task reward, verifier, GRPO loss, and rollout-management strategy unchanged.

Multi-model reasoning and transfer. Drafter–target and role-based multi-agent systems divide inference across models to improve test-time efficiency or reasoning quality (Chu et al., 2025b; Wang et al., 2025). Internal representations can also support inter-model communication and knowledge extraction through hidden states and KV caches, bypassing explicit textual handoffs (Jiao et al., 2024; Fu et al., 2025; Zou et al., 2025; Jiao et al., 2026a). These works study an alternative communication interface for multi-model reasoning. Separately, successful on-policy distillation (OPD) depends on compatibility between teacher and student token distributions (Li et al., 2026). TRLVR pursues a complementary direction: it retains the ordinary token interface and places the frozen weaker partner inside the reward-bearing rollout, so that compatibility is learned as part of the senior's reasoning post-training.

B Reproducibility

B.1 Tandem rollout implementation

A naive realization of tandem rollout couples two HuggingFace models in an outer Python loop with manual KV-cache management. We built such a prototype and found it impractical for RL training: at 512 generated tokens it exhausts the memory of a single 80 GB GPU, making RLVR under long chain-of-thoughts for mathematical reasoning tasks infeasible. Moreover, per-step throughput at shorter lengths is already roughly $4\times$ slower than a single-model vLLM (Kwon, 2025) rollout. Since RL training is dominated by rollout time, this slowdown alone makes large-scale tandem RL infeasible.

We therefore implement the tandem rollout *inside* vLLM, with both π_{sen} and π_{jun} treated as paged-attention-aware models in the same engine. The result is a decoding path that pays only the dual-forward cost (theoretically twice the single-model latency) and inherits every other vLLM optimization: paged KV cache, continuous batching, FlashAttention, CUDA graphs, and tensor parallelism. Empirically our throughput is approximately $0.5\times$ that of a single-model vLLM rollout, matching the architectural lower bound for sequential dual decoding.

B.1.1 Dual-decoder backend

In our implementation, π_{sen} and π_{jun} are loaded onto separate devices and registered in the same vLLM engine, with the junior's layers carrying a fixed name prefix so that the engine's forward context can route them without interfering with the senior's. The two models share per-step attention metadata (slot mappings, block tables, position indices), since they process identical token sequences, but each writes to its own KV-cache tensor on its own device. At each generation step the engine performs one forward pass through π_{sen}, then one through π_{jun} on the same appended token, samples independently from each model's logits, and a tandem sampler emits the chosen token according to the active-model schedule of §3.3; the chosen token is then fed to both models on the next step, keeping the two KV caches in lockstep with the shared response $y_{<t}$. This isolates the tandem-specific

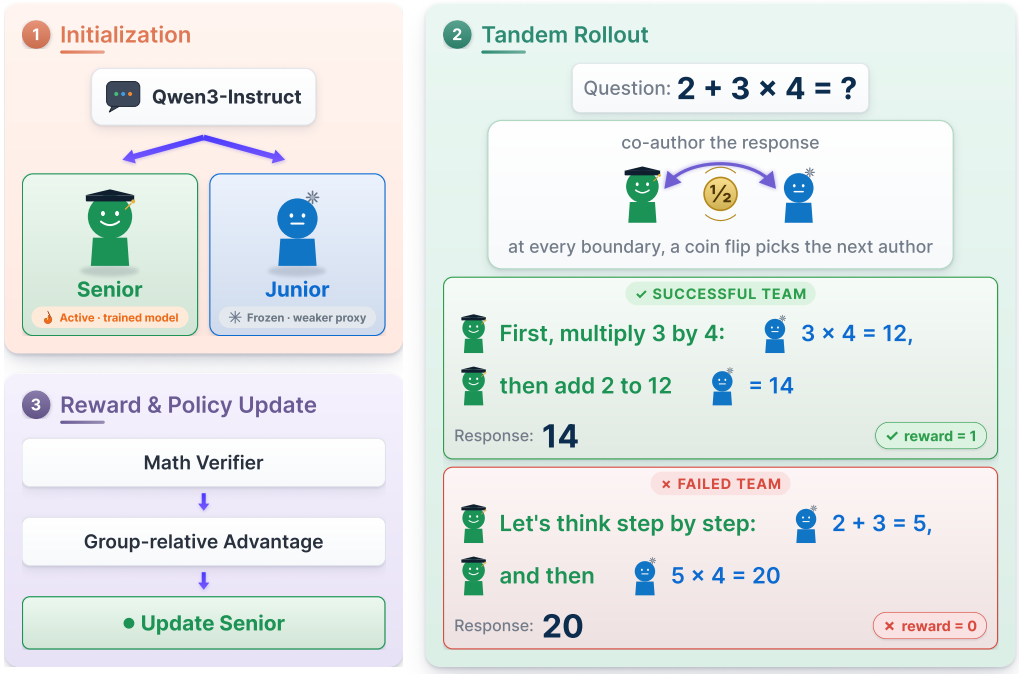

Figure 5: TRLVR at a glance.

logic to model loading, the second forward, and the sampler, while leaving every other vLLM runtime path untouched.

For each request, the engine also accumulates a per-token authorship stream alongside the generated tokens, flagging which positions were emitted by π_{sen}. The stream is propagated through the same output channels as the tokens themselves (model-runner output, engine-core output, per-request completion output) and is exposed to the verl trainer as a tensor field of the rollout batch. On the trainer side, the actor's loss path consumes it by elementwise-multiplying it into the per-token response mask before the policy-gradient sum, which yields exactly the senior-only loss of §3.4. This is the only addition to the verl GRPO surface; the actor, critic, and trainer code paths are otherwise unchanged.

B.1.2 Word-level handoff details

With the vLLM-native backend in place, we now detail the word-level handoff schedule of §3.3 as instantiated in our implementation.

The set of word-boundary tokens is auto-resolved from the active tokenizer at engine initialization. For any BPE tokenizer using a leading-space marker (the Qwen3 / GPT-style word-initial prefix), The boundary set is exactly the set of vocabulary IDs whose surface form begins with that marker. For the Qwen3 tokenizer used in this work, this yields roughly 53k of 151k IDs.

The boundary set alone, however, is not sufficient. Mathematical reasoning produces long runs of non-boundary tokens inside internal LaTeX expressions, variable identifiers, and code blocks, where leading-space tokens never appear. Without intervention, such runs would receive no handoff, allowing the active model to retain control across an entire derivation step and reintroducing exactly the reward-hacking surface that word-level granularity was meant to close. We therefore add a backstop: once K consecutive non-boundary tokens have elapsed since the last redraw, the active model is redrawn under the same Bernoulli(p) rule used at boundaries. We use $K = 32$, chosen as a default that prevented monopoly behaviour in our early rollouts.

The Bernoulli draws at boundary and cap-fallback positions use the per-request random number generator when available and otherwise use PyTorch's global generator. To avoid replaying the full

Table 3: Hyperparameters for training TRLVR on DeepScaleR with Qwen3-4B-Instruct-2507. The GRPO training block matches the vanilla GRPO baseline; the tandem-rollout block lists the additional TRLVR settings.

Category	Hyperparameter
GRPO training	
Learning rate	1×10^{-6}
PPO clip ratio	0.2
Train batch size	16
PPO mini-batch size	8
Group size G (rollouts per prompt)	8
Max response length	3000
Entropy coefficient	0
Advantage estimator	GRPO (group-relative, std-normalised)
Tandem rollout (TRLVR-specific)	
Selection strategy	word-level
Handoff probability p	0.5
Subword-span cap K	32
Junior-token loss weight λ_{jun}	0
Senior π_{sen} initialisation	Qwen3-4B-Instruct
Junior π_{jun} initialisation	Qwen3-4B-Instruct
Generation	
Train temperature	0.6
Train top-p	1.0
Train top-k	-1
Evaluation temperature	0.7
Evaluation top-p	0.8
Evaluation top-k	20

token history at every generation step, the tandem sampler maintains a small per-request state (next active model, tokens since the last redraw) and resumes from this state on each subsequent step.

B.2 DATASETS

We train on DeepScaleR (Tan et al., 2025), a corpus of approximately 40,000 competition-math problems compiled from past olympiad and contest sources. Each example provides a problem statement and a verifiable final answer; reward is computed by extracting the final `\boxed{}` expression from the model's response and comparing it against the gold answer with a sympy-based numeric or symbolic match adopted from Hendrycks et al. (2021). For reproduction, the released data-preparation script holds out 1,000 DeepScaleR problems from training for validation.

For evaluation we use a panel of four competition-math benchmarks, each disjoint from the training corpus:

- **AMC 23–25** [4]: 121 problems.
- **AIME 24–26** [5]: 90 problems.
- **Minerva Math** [6]: 272 problems.
- **OlympiadBench**[7]: 581 olympiad-level problems.

Each problem is formatted with an instruction requiring the model to output its final answer in `\boxed{}` form. We score every rollout with the same boxed-extraction grader used during training;

[4] Compiled from AMC 10/12 problems (American Mathematics Competitions) of 2023, 2024, and 2025.
[5] Compiled from AIME (American Invitational Mathematics Examination) of 2024, 2025, and 2026.
[6] `https://huggingface.co/datasets/math-ai/minervamath`
[7] `https://huggingface.co/datasets/Hothan/OlympiadBench`

aggregate metrics are reported as pass@k over k independent rollouts per problem following the unbiased estimator proposed in Chen et al. (2021).

B.3 Hyperparameter configuration

Table 3 summarises the hyperparameters used to train TRLVR and the vanilla GRPO baseline. All TRLVR experiments reported in this paper were trained on 2× NVIDIA A100 80GB GPUs, with π_{sen} on `cuda:0` and π_{jun} on `cuda:1`; both models run with tensor-parallel size 1.

B.4 End-to-end TRLVR pipeline

Algorithm 1 TRLVR.

Require: Base model π_0; training set $\mathcal{D}$; batch size B; group size G; max response length L; word-boundary set $\mathcal{B}$; subword-span cap K; Bernoulli probability p; verifier $r(\cdot)$

Initialize $\pi_{\text{sen}} \leftarrow \pi_0$ and $\pi_{\text{jun}} \leftarrow \pi_0$; freeze π_{jun}
for $\tau = 1, 2, \ldots$ **do**
  Sample a batch of prompts $\{x_j\}_{j=1}^{B} \sim \mathcal{D}$
  for each (j, i) with $j \in [B]$ and $i \in [G]$ **do**
    // Tandem rollout (§3.3)
    Draw $a_1 \sim \text{Bernoulli}(p)$ over $\{\text{sen}, \text{jun}\}$; $c \leftarrow 0$
    for $t = 1, \ldots, L$ **do**
      Forward π_{sen} and π_{jun} on $x_j \cdot y_{<t}$; sample $y_t^{\text{sen}}, y_t^{\text{jun}}$ independently
      $y_t \leftarrow y_t^{a_t}$; $m_t \leftarrow \mathbb{1}[a_t = \text{sen}]$
      break if y_t is the end-of-sequence token
      if $y_t \in \mathcal{B}$ **or** $c \geq K$ **then**
        Draw $a_{t+1} \sim \text{Bernoulli}(p)$ over $\{\text{sen}, \text{jun}\}$; $c \leftarrow 0$
      else
        $a_{t+1} \leftarrow a_t$; $c \leftarrow c + 1$
      end if
    end for
    Record $y^{(j,i)} \leftarrow y_{1:t}$ and authorship mask $m^{(j,i)} \leftarrow m_{1:t}$
  end for
  // Senior-only GRPO update (§3.4)
  Compute rewards $r^{(j,i)} \leftarrow r(y^{(j,i)})$
  Compute group-relative advantages $A_t^{(j,i)}$ from $\{r^{(j,i)}\}_{i=1}^{G}$
  Update π_{sen} with the GRPO objective of §3.4, with response mask narrowed by $m^{(j,i)}$
end for

C Additional Results

C.1 Statistical significance of TRLVR gains

We complement the bootstrap standard errors of §4 with paired one-sided t-tests on per-problem statistics, mean-pooled across AMC, AIME, and Minerva. KL-Reg here denotes the default $\beta = 10^{-3}$ arm. For pass@k we test per-problem unbiased estimates (Chen et al., 2021); for per-token cross-entropy and distribution overlap α we average per-problem measurements before testing. We note that each test is one-sided in the claimed direction, and asterisks denote significance levels: $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$.

Table 4 reports paired one-sided t-tests for the three directional claims involving TRLVR in the main text. On handoff accuracy (§4.3), TRLVR's lead over both GRPO and KL-Reg holds at every k, with all eight tests significant at $p < 0.05$ and two tests at $p < 0.01$; the consistency across k rules out a single-k artifact. On junior legibility (§4.4.2), TRLVR's per-token cross-entropy is significantly lower than both baselines, confirming that the senior's chain-of-thought is genuinely more predictable to the junior. The distribution-overlap gap is the sharpest: TRLVR beats both baselines at $p < 10^{-100}$.

Table 4: Paired t-tests for the evaluation gains of TRLVR over baselines.

Comparison	Δ	t	p
Handoff Accuracy (pass@k, Δ in percentage points, ↑)			
TRLVR > GRPO, k=1	+1.42	+2.61	0.005^{**}
TRLVR > GRPO, k=2	+1.72	+2.66	0.004^{**}
TRLVR > GRPO, k=4	+1.88	+2.32	0.010^{*}
TRLVR > GRPO, k=8	+2.07	+1.83	0.034^{*}
TRLVR > KL-Reg, k=1	+1.01	+1.73	0.043^{*}
TRLVR > KL-Reg, k=2	+1.24	+1.90	0.029^{*}
TRLVR > KL-Reg, k=4	+1.62	+2.13	0.017^{*}
TRLVR > KL-Reg, k=8	+2.07	+2.14	0.016^{*}
Junior Legibility (per-token cross-entropy, Δ in nats, ↓)			
TRLVR < GRPO	−0.0146	− 9.73	$<10^{-20}$ ***
TRLVR < KL-Reg	−0.0114	−10.94	$<10^{-24}$ ***
Distribution Overlap (α, Δ in percentage points, ↑)			
TRLVR > GRPO	+1.47	+36.16	$<10^{-139}$ ***
TRLVR > KL-Reg	+1.10	+30.22	$<10^{-113}$ ***

C.2 Phrase-level view of the over-emitted vocabulary

Figure 6 extends the case study of §4.4 from tokens to surface phrases, counted per million tokens over the same generations. Each side shows phrases one senior emits at a higher rate than the other, sized by that ratio on a scale shared across the two sides. The reading matches panel (d) of Figure 3: GRPO's side is the furniture of a formatted report, while TRLVR's side is the connective and checking language that carries one step into the next. Several phrases on the TRLVR side, such as `That is,` and `This gives`, are ones GRPO suppresses below the junior's rate while TRLVR keeps them near it, panel (c)'s pull toward the junior in words.

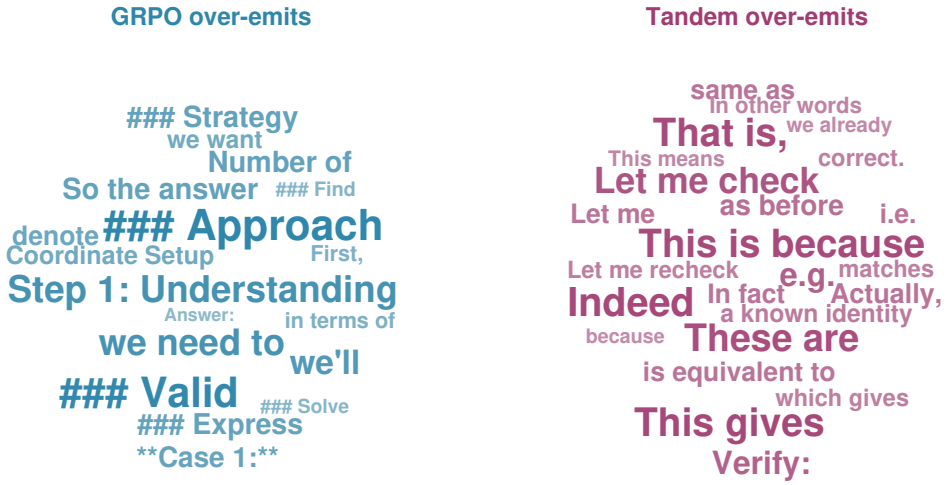

Figure 6: Surface phrases each senior over-emits relative to the other, sized by the rate ratio between the two seniors.

C.3 A step-level case study of junior legibility

The aggregate results in §4.4.2 show that the junior finds TRLVR reasoning less surprising. We next ask whether this advantage is uniform across reasoning steps or concentrated at transitions that rely on prior context. We collect four solo generations per problem from the junior, GRPO senior, and TRLVR senior on 211 AMC and AIME problems, for 2,532 generations in total. A reasoning step is

a span between the double-newline boundaries used in our handoff evaluation. We uniformly sample up to 12 steps from each chain so that long responses do not dominate.

Defining the split. We call a step *self-contained* if, read in isolation, it states what it is doing and makes the symbols and references needed to interpret it identifiable. A step is *context-dependent* if its role or referents require earlier text, for example because it continues a prior calculation or relies on an unresolved pronoun or symbol. To apply this definition, a frozen Qwen3.6-35B judge sees only the current step, without the problem or preceding chain of thought, and is constrained to answer yes or no. We map yes to self-contained and no to context-dependent. The judge prompt is fixed throughout, and the judge is not told which model produced the step.

This isolation is deliberate: it tests whether the step itself carries enough information to be interpreted, rather than whether external context can make it interpretable. Showing the problem or earlier reasoning could supply precisely the information whose absence the split is intended to detect. This is not meant to reproduce the junior's information at inference; we restore the full problem and reasoning prefix when measuring junior surprisal below.

An example pair from one evaluated TRLVR generation.
Self-contained: "Each game has one winner and one loser, so the total number of wins is equal to the total number of games."
Context-dependent: "So $L = 3$ works."
The first states both the premise and its implication locally. The second requires earlier text to identify what L denotes, which condition is being tested, and why 3 satisfies it.

As a model-free robustness check, we also construct a regex split. After trimming whitespace, a step is marked context-dependent if its first word is one of ten fixed context-carrying terms or its final character is a colon; all other steps are marked self-contained.[8]

Comparison. Within each group, we compare how readily the frozen junior predicts the opening of a senior's step, conditioned on the original problem and all preceding reasoning. Table 5 reports the difference in junior cross-entropy, $\Delta\mathrm{CE} = \mathrm{CE}_{TRLVR} - \mathrm{CE}_{\mathrm{GRPO}}$, so negative values indicate lower junior surprisal under TRLVR. Crucially, neither split has access to these likelihoods: *context-dependent* describes reliance on prior text, not whether the junior finds the step difficult.

Table 5: Context-dependent step frequencies and the TRLVR–GRPO gap in junior cross-entropy at step entry under the LLM judge and regex splits.

Split	Context-dependent steps (%)			ΔCE (nats, $\downarrow$)	
	Junior	GRPO	TRLVR	Self-contained	Context-dependent
LLM judge	37.6	35.5	33.5	$+0.0024$	-0.0707
Regex split	41.7	39.3	37.1	$+0.0055$	-0.0667

Results. The self-contained groups exhibit near-zero TRLVR–GRPO gaps under both splits. By contrast, the context-dependent groups show reductions of 0.067–0.071 nats in junior cross-entropy for TRLVR. The splits agree on only 73.5% of steps, yet recover nearly identical conditional gaps. TRLVR also produces slightly fewer context-dependent steps, but prevalence alone does not explain the difference observed within that group.

Because self-containedness is an operational distinction and the semantic split is determined by an LLM judge rather than human annotation, we treat this analysis as a descriptive case study, not a causal account or a universal partition of reasoning steps. Across both splits, the observed junior-legibility difference is concentrated at transitions whose interpretation leans on prior context.

[8]The case-insensitive word list is `so`, `then`, `thus`, `this`, `it`, `therefore`, `now`, `and`, `which`, and `substituting`.

C.4 CONTROLS FOR JUNIOR CROSS-ENTROPY

We use two controls to test whether junior cross-entropy is related to task success and whether its reduction under TRLVR can be explained by the predictability of TRLVR traces under the senior itself.

C.4.1 RELATION TO TASK SUCCESS

We first test whether lower junior cross-entropy is associated with a trace's correctness. We pair every trace used in Table 1 with its binary correctness reward. For each senior, we retain problems containing both correct and incorrect traces among the eight solo rollout samples and compute the junior cross-entropy of correct traces minus that of incorrect traces within each problem. Negative values mean successful traces are less surprising to the junior. We average these differences over problems and obtain 95% confidence intervals by bootstrapping problems.

Table 6: Difference in junior cross-entropy between correct and incorrect senior traces within the same problem (nats, ↓).

Senior	Mixed problems	ΔCE	95% CI
GRPO	239	-0.0027	$[-0.0060, +0.0005]$
TRLVR	242	-0.0078	$[-0.0112, -0.0044]$

Pooling the two seniors, the within-problem difference between correct and incorrect traces is -0.0053 nats, with a 95% confidence interval of $[-0.0079, -0.0026]$. The point estimate is more negative for TRLVR, and only its interval excludes zero. Thus lower junior surprisal is associated with task success on these solo traces. This comparison does not identify a causal mechanism or establish semantic understanding.

C.4.2 CROSS-ENTROPY UNDER THE SENIOR

We next test whether TRLVR traces are simply easier to predict under the model that generated them. We score each trace under its senior and report the difference $\mathrm{CE}_{\mathrm{jun}} - \mathrm{CE}_{\mathrm{sen}}$ on the same observed tokens. If general predictability explained the lower junior cross-entropy, TRLVR would also have lower senior cross-entropy and its advantage would shrink after this subtraction.

Table 7: Junior and senior cross-entropy and their difference on senior traces, reported as a macro average across four benchmarks (nats).

Senior	Junior CE	Senior CE	$\mathrm{CE}_{\mathrm{jun}} - \mathrm{CE}_{\mathrm{sen}}$
GRPO	0.1294	0.0636	0.0658
TRLVR	0.1183	0.0882	0.0300

The TRLVR traces have 0.0247 nats higher senior cross-entropy than the GRPO traces, with a 95% confidence interval of $[0.0233, 0.0261]$. The gap between junior and senior cross-entropy is nevertheless 0.0358 nats lower, with a 95% confidence interval of $[-0.0373, -0.0344]$. This is a 54% reduction in the macro average, and the gap is smaller on all four benchmarks. Under the same decoding settings, this pattern is consistent with GRPO concentrating probability on a narrower set of continuations as it drifts from the junior, whereas TRLVR retains more uncertainty and stays closer to the junior. Thus TRLVR's lower junior cross-entropy is not explained by greater predictability under the generating senior. Because the traces were sampled with temperature and top-p, the gap is a likelihood contrast on sampled traces rather than an exact KL divergence.

C.5 STOCHASTIC REASONING-STEP HANDOFF ACCURACY

The main HR result in §4.3 alternates the senior and the junior strictly at every `\n\n` boundary. This deterministic round-robin is a clean evaluation protocol, but in scenarios such as agent-to-agent collaboration, drafter-target inference, and human-AI handoffs, the timing of handoffs is normally

more flexible and the next contributor is typically chosen by the situation rather than by a fixed cadence. As a robustness check, we replace the deterministic round-robin with an independent Bernoulli($p = 0.5$) draw at every `\n\n` boundary, keeping every other setting identical. The expected senior share remains 0.5, but the per-rollout schedule now varies, and a team's success has to be robust to that variation.

Table 8: Reasoning-step handoff accuracy under stochastic Bernoulli ($p = 0.5$) schedules at `\n\n` boundaries (pass@8, %, ↑). Junior is the Qwen3-4B-Instruct training partner.

Benchmark	GRPO	TRLVR	Δ
AMC	78.5	80.2	+1.7
AIME	38.9	42.2	+3.3
Minerva	44.5	44.9	+0.4
Olympiad	64.0	65.1	+1.1
Macro Avg	56.5	58.1	+1.6

Per-rollout senior shares are well-controlled across methods (mean 50.0% for both, standard deviation in the 5%–9% range across benchmarks). TRLVR retains a directional advantage over vanilla GRPO on every benchmark and on macro, with the largest gap again on AIME ($+3.3$ at pass@8). The macro gap is smaller than under the strict schedule of §4.3 ($+1.6$ vs. $+3.1$), as expected: under a randomised schedule, individual rollouts vary in how much of the trajectory each model carries, and pass@8 aggregates over schedule variation in addition to the usual generation variation. The strict-alternation protocol of the main text is the more stringent measurement; the stochastic protocol here verifies that TRLVR's compatibility lift is not an artefact of a fixed schedule.

D Different training signals in KL regularization and tandem rollouts

A natural alternative to TRLVR is to keep solo GRPO rollouts and add a per-token KL penalty toward the frozen junior π_{jun}. This likewise "anchors" the senior toward the junior, but as a soft additive regularizer on the senior's own trajectories rather than by sampling rollouts from a tandem mixture. Although both objectives involve the same frozen junior, they expose the senior to different training signals. We make this distinction explicit below and relate it to the empirical pattern in our ablation (§5.2).

D.1 Setup and what each objective optimizes

Let $\mathcal{V}$ be the vocabulary, x a prompt, $y = y_{1:T} \in \mathcal{V}^T$ a response, $h_t = x \cdot y_{<t}$ the context at position t, and $r : \mathcal{V}^T \to [0, 1]$ a bounded verifier reward. Both π_θ (senior, trainable) and π_{jun} (junior, frozen) are autoregressive: $\pi(y \mid x) = \prod_{t=1}^{T} \pi(y_t \mid h_t)$. We compare two objectives.

(a) GRPO with KL regularization toward π_{jun}.

$$\mathcal{L}_{\text{KL-Reg}}(\theta) \;=\; -\,\mathbb{E}_{y\sim\pi_\theta}\Big[\textstyle\sum_t A_t \log \pi_\theta(y_t \mid h_t)\Big] \;+\; \beta \cdot \mathbb{E}_{y\sim\pi_\theta}\Big[\textstyle\sum_t \mathrm{KL}\big(\pi_\theta(\cdot \mid h_t) \,\big\|\, \pi_{\text{jun}}(\cdot \mid h_t)\big)\Big]. \tag{2}$$

Rollouts are sampled solo from π_θ; the regularizer penalizes per-position next-token disagreement with π_{jun} along senior-natural trajectories. In implementation, we use verl's low-variance KL estimator (`low_var_kl`), with β set by `kl_loss_coef`.

(b) TRLVR. Let M_p denote the tandem rollout distribution: at each handoff boundary an independent $\mathrm{Bernoulli}(p)$ draw determines the active model $a_t \in \{\text{sen}, \text{jun}\}$, and $y_t \sim \pi_{a_t}(\cdot \mid h_t)$.

$$\mathcal{L}_{\text{TRLVR}}(\theta) \;=\; -\,\mathbb{E}_{y\sim M_p}\Big[\textstyle\sum_{t\,:\,a_t=\text{sen}} A_t \log \pi_\theta(y_t \mid h_t)\Big]. \tag{3}$$

Since only senior-emitted positions depend on θ, the standard policy-gradient identity gives

$$\nabla_\theta\, \mathbb{E}_{y\sim M_p}[r(y)] \;=\; \mathbb{E}_{y\sim M_p}\Big[r(y) \textstyle\sum_{t\,:\,a_t=\text{sen}} \nabla_\theta \log \pi_\theta(y_t \mid h_t)\Big]. \tag{4}$$

Unlike KL-Reg, TRLVR evaluates the reward on joint rollouts sampled under the tandem mixture schedule. Crucially, r is evaluated on a trajectory that includes junior-emitted tokens, so the team-reward feedback at any senior-emitted position t reflects whether the junior's contributions remain compatible with the rest of the trajectory. KL-Reg, as instantiated here, receives the senior's solo reward and a token-level divergence penalty, but no task reward from trajectories in which the junior participates. This is the additional training signal introduced by tandem rollouts. Consistent with this distinction, among the coefficients evaluated in our ablation, KL-Reg either retains the communication tax of vanilla GRPO or improves junior legibility at the cost of solo and team performance. TRLVR is the only evaluated arm that preserves GRPO-level solo capability while retaining 99.8% of its own capability under handoff.