# VGAC: Verifier-Grounded Agreement-Calibrated Reinforcement Learning for Long-Horizon Agents

A Research Proposal for Reliable, Sample-Efficient Agent Training

Vyomakesh Dundigalla

March 2026

**Abstract**

Reinforcement learning with verifiable rewards (RLVR) gives large language model agents a clean outcome signal, but it often gives weak credit assignment in long-horizon settings. An agent can be confidently wrong for many steps before the final verifier exposes failure, which wastes rollouts and makes quiet errors difficult to correct. This proposal studies whether *on-policy internal confidence signals*, grounded by verifier outcomes, can improve both training-time updates and inference-time control for long-horizon agents. The core thesis is simple: **verifier-grounded agreement across grouped rollouts can improve long-horizon RL agents primarily by reducing silent overconfident failures**. The core idea is to use grouped rollouts, sequence entropy, and agreement across rollouts to derive a verifier-grounded confidence score that reshapes reward and triggers verify/replan when confidence is low. The proposal is deliberately narrower than generic uncertainty-aware RL: silent-failure reduction is the primary claim, while sample efficiency and retention are secondary hypotheses. We outline a baseline RLVR setup, a calibration-aware reward shaping method, a retention-aware training protocol, and an experimental plan spanning signal studies, long-horizon agent tasks, and catastrophic forgetting evaluation.

## 1 Introduction

Long-horizon agents are attractive because they can plan, use tools, recover from intermediate mistakes, and solve tasks that are too extended for single-shot prompting. They are also brittle. Small early errors compound, and a trajectory that looks locally plausible can drift for many steps before the final verifier marks the entire episode as wrong. In standard RLVR, that failure is informative at the episode level but not necessarily at the level of *how sure the model should have been*, *when it should have verified*, or *when it should have replanned*.

This proposal is motivated by a simple observation: outcome-only reward is often too sparse for reliable long-horizon behavior. If an agent can express or expose a grounded notion of uncertainty, then that signal can be used in two complementary places. First, it can improve *training-time credit assignment* by penalizing overconfident failures more strongly than low-confidence exploratory failures. Second, it can improve *inference-time control* by deciding when to continue, when to verify, and when to replan.

The proposal does *not* claim that "confidence" is a new idea. Recent work has already explored answer calibration, entropy-based stopping, uncertainty-aware rewards, and overconfidence penalties. What remains underexplored is a clean, on-policy, verifier-grounded way to use confidence in long-horizon RL agent loops without turning the method into frontier-model distillation. The thesis is intentionally narrow: **verifier-grounded agreement across grouped rollouts can improve long-horizon RL agents primarily by reducing silent overconfident failures**. Entropy and verifier partial scores

support that signal, but silent-failure reduction is the headline claim; sample efficiency and retention are secondary hypotheses.

The proposal makes four modest contributions:

- a verifier-grounded confidence signal derived from grouped rollouts rather than a frontier teacher;
- an on-policy calibration-aware reward for long-horizon RLVR agents;
- a clear separation between training-time shaping and inference-time verify/replan behavior;
- a retention-aware evaluation protocol that treats catastrophic forgetting as a measured risk rather than an afterthought.

The direction is also strategically aligned with the broader research agenda around calibration-aware reasoning, long-horizon agents, and efficient experimentation. Recent work in this area has emphasized whether models know when they are correct and how confidence can save compute at inference time. This proposal is tailored to the next step in that agenda: turning verifier-grounded confidence from an inference-time diagnostic into a training-time credit-assignment signal for agent loops. The execution plan is intentionally small-first and intern-realistic: start with a Qwen-class 3B–7B policy, grouped rollouts with $K \in \{4, 6\}$, one short verifiable domain, and one long-horizon domain with reliable verifier feedback; only scale after the signal study shows that agreement adds predictive power beyond entropy. A successful pilot would already be valuable because it would produce reusable infrastructure for silent-failure measurement, verifier-routed control, and a clearer answer to which confidence signals are worth trusting.

## 2 Related Work and the Remaining Gap

The closest literature falls into five clusters. First, calibration papers such as Stangel et al. [4] and Xu et al. [6] teach models to express calibrated confidence, mostly at the answer level. Second, reasoning-focused calibration work such as Deng et al. [1] studies calibration in multi-step reasoning but does not define a full agent-training loop. Third, entropy and uncertainty are increasingly used at inference time for adaptive control, including entropy-based stopping [3] and training-free uncertainty-aware agent frameworks [11]. Fourth, RL and RLHF work such as Zhai et al. [8] inject uncertainty into reward modeling to reduce overoptimization. Fifth, several recent agent papers directly touch the neighborhood of this proposal: entropy-modulated policy updates [5], uncertainty-aware rewards [10], confidence-shaped RL rewards [9], calibrated process reward models [2], and asymmetric penalties for overconfident errors [7].

This makes the broad space crowded. A proposal that simply says "use confidence in RL" is no longer differentiated. Three areas in particular already look crowded:

- generic uncertainty-aware reward shaping;
- entropy-only confidence estimation;
- external frontier models acting as judges of a smaller model's confidence.

The remaining gap is narrower: **on-policy, verifier-grounded, agreement-based calibration for long-horizon RL agents, with explicit attention to silent failure, verify/replan decisions, and sample efficiency**. That gap matters because grouped rollouts naturally expose a new kind of internal evidence: if several low-temperature rollouts converge to the same answer, that agreement is often more informative than a model's self-reported confidence. At the same time, agreement must still be grounded by a verifier, because all rollouts can be confidently wrong.

| Work | Signal source | Train / infer | Long-horizon | Verifier-grounded | Positioning relative to this proposal |
|---|---|---|---|---|---|
| Rewarding Doubt [4] | model confidence | train | no | no | Calibrated confidence expression, mostly answer-level |
| SaySelf [6] | self-reflective rationale | train | no | no | Confidence expression with RL, not long-horizon agent control |
| Unified View [1] | calibration analysis | infer | partly | no | Useful framing for step/path calibration, not an RL agent loop |
| UP-RLHF [8] | reward-model uncertainty | train | no | no | Uncertainty in RLHF, but not agent self-calibration |
| Think Just Enough [3] | sequence entropy | infer | no | no | Early stopping / compute control at inference time |
| PRM Calibration [2] | calibrated PRM uncertainty | infer | partly | yes | Strong prior for verifier-side calibration, not on-policy rollout agreement |
| EMPG [5] | entropy | train | yes | no | Very close prior on uncertainty-modulated updates |
| SELAUR [10] | uncertainty-aware reward | train | yes | partly | Very close prior on reward shaping for multi-step agents |
| ConfClip [9] | confidence-weighted reward | train | partly | partly | Adjacent reward shaping prior |
| ACE [7] | overconfidence penalty | train | partly | no | Strong prior on punishing overconfident errors |
| This proposal | entropy + rollout agreement + verifier score | both | yes | yes | On-policy, verifier-grounded calibration focused on silent failure and verify/replan |

Table 1: Positioning the proposal against the closest related work.

# 3 Method

## 3.1 Baseline RLVR setup

We assume a single open policy family for both baseline and treatment conditions, ideally a Qwen-class model in the 3B–7B range for the pilot. A concrete first implementation would use a Qwen3-4B class policy with verifier-compatible tasks already close to PRIME-RL workflows: GSM8K plus a filtered MATH subset for short verifiable reasoning, then Wiki Search as the first long-horizon tool-use domain, with scientific or molecular-design tasks that expose reliable verifier signals as a more ambitious follow-on setting. For each prompt $x$, the policy $\pi_\theta$ produces $K$ rollouts,

$$y_1, \ldots, y_K \sim \pi_\theta(\cdot \mid x),$$

and each rollout receives a task reward $r_i \in [0, 1]$ from a verifier. The baseline uses standard grouped RLVR / GRPO-style updates, where the policy improves according to relative episode reward but has no explicit notion of calibrated confidence. Throughout the proposal, *on-policy* refers to how the confidence signal is constructed: from the current policy family's own grouped rollouts rather than from an external teacher model. A practical PRIME-RL implementation would still be mildly off-policy at the optimizer level, because asynchronous inference may sample from a slightly stale policy and train with a token-level AIPO-style objective. The intended execution should therefore be read as *on-policy signal construction with asynchronous off-policy optimization*, not as a claim of perfectly synchronous on-policy training.

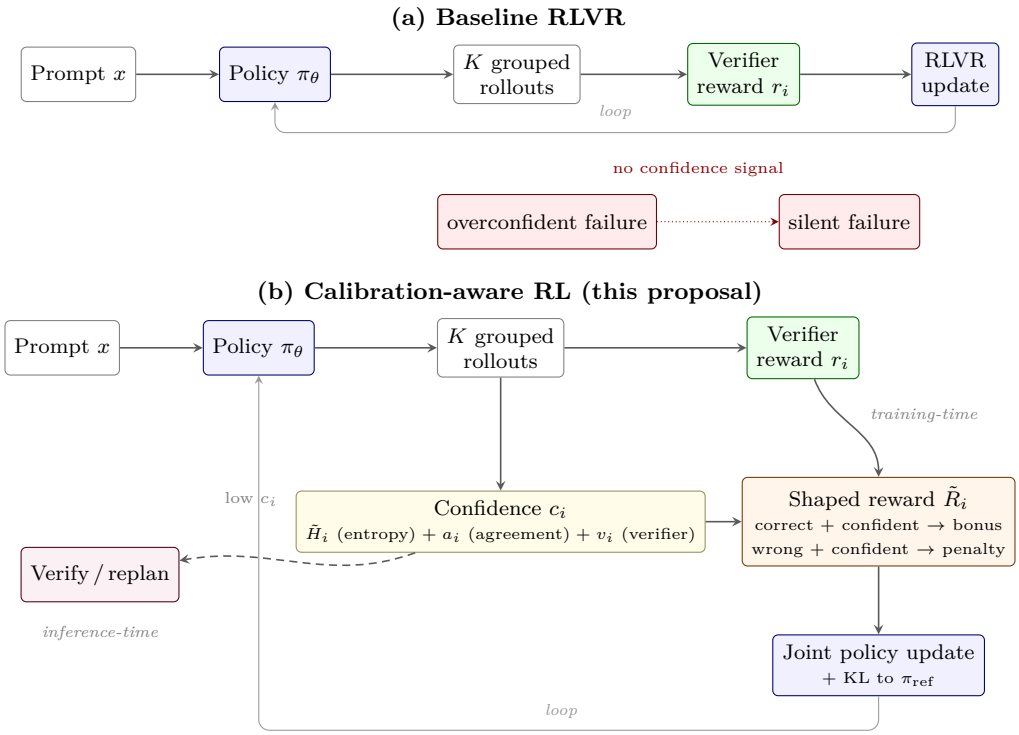


Figure 1: Baseline RLVR versus the proposed calibration-aware loop. The treatment arm uses an on-policy confidence signal for *training-time* reward shaping (solid right path) and *inference-time* verify/replan routing (dashed left path).

### 3.2 Verifier-grounded confidence without a trainable head

The first version of the method intentionally avoids a trainable confidence head. A trainable head is tempting, but without a grounded target it is easy to reward-hack or collapse. Instead, V1 derives confidence from three sources:

1. **normalized sequence entropy**, which measures how diffuse the next-token distributions are along the trajectory;
2. **rollout agreement**, which measures whether grouped rollouts converge to the same final answer under controlled sampling;
3. **verifier partial score**, when the verifier returns something richer than binary correctness.

For rollout $y_i$ with token length $T_i$, define average sequence entropy

$$H_i = -\frac{1}{T_i} \sum_{t=1}^{T_i} \sum_{v} p_t(v) \log p_t(v),$$

where $p_t(v)$ is the next-token distribution at step $t$. Let $\tilde{H}_i$ denote a normalized entropy score, and define answer agreement

$$a_i = \frac{1}{K-1} \sum_{j \neq i} \mathbf{1}[\text{ans}(y_i) = \text{ans}(y_j)].$$

If the task has a partial verifier signal $v_i \in [0,1]$, we combine these into a derived confidence score:

$$c_i = \sigma\Big(\alpha(1-\tilde{H}_i) + \beta a_i + \gamma v_i\Big).$$

This score is *internal* in the sense that it is generated from the policy's own grouped behavior, but it is *grounded* because agreement alone is never treated as correctness.

These coefficients are not claimed to be theoretically optimal. In V1, the proposal treats them as calibrated design parameters: normalize $(1 - \tilde{H}_i)$, $a_i$, and $v_i$ to a shared scale, initialize $\alpha = \beta = \gamma = 1$, then tune them on held-out Phase 1 data using simple grid search or a logistic-regression fit to maximize AUROC and minimize Brier score. Sensitivity sweeps around $\{0, 0.5, 1, 2\}$ should be part of the first ablation pass. This matters especially in the hardest edge case: when all $K$ rollouts agree on the wrong answer, $a_i$ can be spuriously high, so the verifier-side term $\gamma v_i$ becomes load-bearing in pulling $c_i$ back down.

Agreement matters because it captures a different failure mode than entropy. Entropy is a *local* signal: it tells us whether the token distribution is diffuse at a given step, but a trajectory can still have low entropy while converging confidently to a wrong answer. Agreement is a *cross-trajectory* signal: it asks whether independently sampled rollouts stabilize on the same solution. In that sense, entropy measures local uncertainty while agreement measures global solution stability. The two are complementary rather than redundant, and Phase 1 is designed to test that claim directly through entropy-only, agreement-only, and combined ablations. Verifier grounding remains necessary because high agreement can still arise in the all-rollouts-wrong case.

### 3.3 Calibration-aware reward shaping

Let

$$z_i = \mathbf{1}[r_i \geq \tau]$$

indicate whether the rollout passes a correctness threshold. We define a calibration-aware shaped reward

$$\tilde{R}_i = r_i + \lambda_b z_i c_i - \lambda_p (1 - z_i) c_i.$$

Here $\lambda_b$ is the confidence bonus coefficient and $\lambda_p$ is the confidence penalty coefficient. Read left-to-right, this says: start with verifier reward $r_i$, add a bonus for confident correct rollouts, and subtract a penalty for confident wrong rollouts. The first extra term rewards confidence when the rollout is correct. The second penalizes confidence when the rollout is wrong. This is the core idea: *overconfident failures should be corrected more strongly than low-confidence exploratory failures.* Group-relative advantages are then

$$A_i = \tilde{R}_i - \frac{1}{K} \sum_{j=1}^{K} \tilde{R}_j,$$

and the policy update uses a KL-regularized objective

$$L_{\text{policy}} = -\mathbb{E}\big[ \log \pi_\theta(y_i \mid x) A_i \big] + \beta \, \mathrm{KL}(\pi_\theta \,\|\, \pi_{\text{ref}}).$$

The reference policy $\pi_{\text{ref}}$ stabilizes learning and also helps preserve prior ability. In an actual PRIME-RL training stack, this shaped reward would be mapped into token-level advantages under the async AIPO-style trainer rather than replace the trainer with a purely episode-level loss.

### 3.4 Inference-time verify / replan policy

The training-time reward does not fully specify how the agent should behave at deployment. At inference time, the proposal uses confidence as a routing signal:

- if confidence is sufficiently high, continue the trajectory;

- if confidence is low on a hard step, trigger *verify* or *replan*;
- if the verifier indicates likely failure, treat confidence as zero and force correction.

This keeps the roles clean: verifier outcomes remain the ground-truth signal, while confidence determines how aggressively the system should spend extra compute or invoke tools.

### 3.5 Optional V2: a trainable confidence head

Only after V1 is established should a trainable head be introduced. In V2, a lightweight head $c_\phi(\cdot)$ predicts correctness from a representation built from the rollout's last-layer hidden states and V1 signals. A reasonable first design is a two-layer MLP on top of a pooled hidden-state summary $h_i^{\text{pool}}$ concatenated with scalar features $[(1-\tilde{H}_i), a_i, v_i]$, producing a single probability $\hat{c}_i \in [0,1]$. The target remains verifier-grounded:

$$L_{\text{cal}} = \text{BCE}(\hat{c}_i, z_i)$$

or alternatively a Brier loss if sharp calibration is prioritized. The first V2 ablation should keep the calibration loss on detached features or a frozen backbone so that the head learns to read confidence without destabilizing the main policy; a second ablation can allow end-to-end gradients through the backbone. In practice, the combined objective would be

$$L_{\text{total}} = L_{\text{policy}} + \lambda_{\text{cal}} L_{\text{cal}},$$

with $\lambda_{\text{cal}}$ tuned on Phase 1 and Phase 2 validation data. The proposal must remain explicit that the confidence head is optional because its value must be demonstrated against the simpler V1 signal stack, not assumed.

## 4 What Stays In the Architecture, and What Stays Out

Several design choices are intentional. Prime verifiers stay in the loop because confidence without grounding is not enough. Grouped rollouts stay in the loop because agreement is the cleanest on-policy evidence that a model has found a stable answer. Entropy stays in the loop as a secondary signal because it captures local uncertainty, even if it is not sufficient by itself. Verify/replan stays in the loop because calibration is only useful if it changes behavior. KL regularization stays in the loop because the proposal cares about retention and stability, not just gains on one target domain.

Three choices are removed from the core claim. First, an external GPT-style model should *not* be the main confidence scorer. That makes the method teacher-centered and slides it toward distillation. Second, pure self-reported confidence should not be trusted as the primary signal; it is too easy to game. Third, a confidence head without grounded targets should not be introduced in V1.

What seems to work in the literature is narrow and concrete: grouped rollouts, verifier-grounded labels, uncertainty-modulated updates, and targeted inference-time verification. What seems less reliable is entropy alone, open-ended self-reported confidence, and any setup in which all confidence supervision comes from a stronger external model. A particularly important failure mode is the *all-rollouts-wrong* case: high agreement across six wrong rollouts is not confidence, it is a coordinated failure. That is exactly why verifier grounding is required.

# 5 Experimental Design

## 5.1 Strategic alignment and practical execution

The proposal is designed to support a concrete research program rather than float beside it. It targets long-horizon agent reliability, calibration-aware control, and sample-efficient experimentation, all of which are already visible themes in current agent-reliability work. Just as importantly, the design is practical. The core loop depends only on components that fit a realistic pilot: a small open policy model, grouped rollouts through Prime RL Labs, Prime verifiers as the sole correctness oracle, and environments already compatible with `oai_tools`. The first milestone is not "train a frontier agent"; it is to establish whether agreement predicts correctness beyond entropy on one short verifiable domain and one long-horizon domain. The second milestone is to show a reduction in silent failure at comparable task success. Only after those gates are met does it make sense to test a confidence head or larger models. This sequencing keeps the proposal resource-aware while still aiming at a reusable research result.

A concrete pilot would use GSM8K and a filtered MATH slice (for example, the Hendrycks Sanity-style 20–80% solvable regime) as short verifiable domains, plus Wiki Search as the first long-horizon tool-use benchmark, with scientific or molecular-design tasks that provide reliable verifier feedback as a more ambitious follow-on. With a Qwen3-4B class model, $K = 6$ rollouts, roughly 5k short-domain prompts, 1k–2k long-horizon episodes, and 128–256 generated tokens per rollout, the Phase 1 signal study is on the order of 4–8M generated tokens. On 2–4 H100-class GPUs or equivalent async inference/training capacity, that is a pilot in the tens of GPU-hours for signal extraction and the low hundreds of GPU-hours for the full baseline-versus-treatment ablation suite, which is large enough to be meaningful but still reasonable for an Research project.

## 5.2 Hypotheses

The main hypotheses are:

1. **H1**: rollout agreement predicts correctness better than entropy alone;
2. **H2**: calibration-aware reward reduces silent failure rate at equal or better task success;
3. **H3**: confidence-gated verify/replan improves success per verifier call and per token or step;
4. **H4**: calibration-aware shaping improves sample efficiency over verifier-only RLVR;
5. **H5**: replay, KL regularization, and adapter-based tuning preserve prior-task performance better than naive fine-tuning.

## 5.3 Phases

The study should proceed in four phases.

**Phase 1: signal study without RL updates.** Compare entropy, rollout agreement, verifier partial scores, and optional self-reported confidence as predictors of correctness. This phase answers the strongest question: *what actually measures confidence well enough to use in training?*

**Phase 2: RL baseline versus calibration-aware shaping.** Keep the policy family, verifier, and task family fixed. Compare verifier-only RLVR against entropy-only, agreement-only, and combined calibration-aware reward shaping.

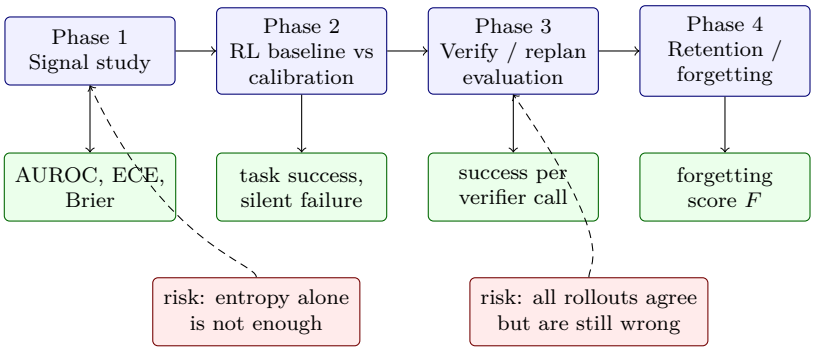


Figure 2: Phased evaluation plan. The proposal is intentionally staged so that confidence signals are validated before they are promoted to policy-shaping signals.

**Phase 3: inference-time verify/replan.** Add a control policy that spends extra compute or verifier calls only when confidence falls below threshold. Evaluate cost-success trade-offs rather than raw success alone.

**Phase 4: retention and forgetting.** Train on target tasks while periodically evaluating on anchor tasks. This keeps the proposal honest about catastrophic forgetting.

### 5.4 Tasks, ablations, and metrics

The task mix should include one short verifiable domain, one long-horizon agentic domain, and one mixed retention protocol. A practical configuration is:

- short verifiable reasoning: GSM8K plus a filtered MATH subset with answer-based verification;
- long-horizon agentic setting: Wiki Search as the first multi-turn tool-use benchmark, followed by scientific or molecular-design tasks with reliable verifiers if the signal study is stable;
- retention protocol: anchor tasks from GSM8K / MATH plus target long-horizon tasks from Wiki Search or a scientific / molecular-design domain with reliable verifier feedback.

Ablations should include:

- verifier-only RLVR;
- entropy-only confidence;
- agreement-only confidence;
- entropy + agreement;
- with and without verifier partial scores;
- with and without inference-time verify/replan;
- optional external-model teacher as ablation only;
- optional V2 confidence head.

Implementation-specific ablations should also be included once the method is wired into PRIME-RL: verifier-only AIPO versus AIPO with calibration-aware shaping, async gap sensitivity through `max_async_level`, reward-to-token credit assignment choices, and sensitivity to clipping / KL coeffi-

cients. These are important because several recent RL systems papers have shown that apparently small loss-design choices can materially affect stability.

Core metrics are:

- task success and pass@1;
- silent failure rate;
- recovery rate after bad intermediate steps;
- verifier usage rate;
- expected calibration error (ECE), Brier score, and AUROC for correctness prediction;
- token cost, step cost, and cost per correct decision.

We define *silent failure rate* operationally as the fraction of failed trajectories that remained high-confidence until failure without triggering verification or replanning. For rollout $y_i$, let $f_i = \mathbf{1}[r_i < \tau]$ denote verifier failure, let $c_i$ be the derived confidence score, and let $u_i \in \{0, 1\}$ indicate that no verify/replan trigger fired before failure. For a confidence threshold $\eta$, we measure

$$\mathrm{SFR} = \frac{\sum_i \mathbf{1}[f_i = 1 \wedge c_i \geq \eta \wedge u_i = 1]}{\sum_i \mathbf{1}[f_i = 1]}.$$

In words, the numerator counts failed rollouts that were still above the confidence threshold and never triggered verify/replan, while the denominator counts all failed rollouts. This makes the main claim testable: the treatment arm should lower silent failure even when overall success changes only modestly.

## 6 Long-Horizon Agents, Continual Learning, and Sample Efficiency

This proposal does not claim to solve continual learning outright. That would be too broad. Instead, it treats continual learning as a *secondary evaluation axis* and asks whether calibration-aware RL can improve target-task behavior *without unacceptable forgetting*. The retention controls are simple:

- mixed-task training batches;
- a replay buffer of anchor tasks;
- KL regularization to a frozen reference policy;
- adapter-based fine-tuning (for example, LoRA or other PEFT methods) for pilot runs instead of unrestricted full-model drift.

To quantify forgetting, let $S_t^{\mathrm{before}}$ and $S_t^{\mathrm{after}}$ denote performance on anchor task $t$ before and after calibration-aware training. We measure

$$F = \frac{1}{N} \sum_{t=1}^{N} \max\left(0, S_t^{\mathrm{before}} - S_t^{\mathrm{after}}\right).$$

The method helps long-horizon agents because it gives earlier warning that a trajectory is drifting, and it gives the learner a stronger penalty for being confidently wrong. It helps sample efficiency because failed rollouts are no longer almost-binary dead ends; they still carry structured calibration signal. It may help continual learning indirectly because KL regularization, replay, and PEFT make it easier to preserve prior skills while studying the effect of the new reward. But this should be written as a measured hypothesis, not as an assumption of success.

# 7 Risks, Failure Modes, and Practical Boundaries

There are several realistic failure modes. The model could learn to stay low-confidence everywhere, which would reduce overconfident mistakes but also destroy usefulness. The verifier could be noisy or overly sparse, which would weaken the grounding of the calibration signal. Grouped rollouts could agree for the wrong reasons, especially on biased or under-specified tasks. A trainable confidence head could reward-hack unless its targets are firmly tied to verifier outcomes. And a frontier teacher added to the loop could make the method look like distillation instead of on-policy calibration learning.

These risks suggest practical boundaries. The first proposal should be scoped to small open models, verifiable tasks, and grouped RLVR. The proposal should also be explicit that external models are optional baselines or weak-teacher ablations, not the main method. Finally, negative results would still be informative: if agreement-based confidence fails to outperform entropy, that would say something important about what signals are worth trusting in long-horizon agent training.

# 8 Conclusion

The case for this proposal is not that confidence is new. It is that long-horizon RL agents still need a better way to tell the difference between exploratory failure, recoverable uncertainty, and silently overconfident error. A verifier-grounded, on-policy confidence signal built from grouped rollouts offers a concrete path forward. If it works, the gains should appear not only in final success, but in lower silent failure, better verify/replan behavior, and more value extracted from failed trajectories. That is a narrow enough claim to be defensible, and useful enough to matter.

# A Appendix: FAQ and Direct Responses to Likely Questions

## A.1 Signal Roles and Their Uses

| Signal type | What it answers | Primary use in the proposal |
|---|---|---|
| Prime verifier reward | "Was this output good?" "Did the agent solve the task?" "How correct was this trajectory or result?" | Ground-truth outcome signal for reward shaping, evaluation, and correctness labels |
| Calibration signal | "Did the model know it might be wrong?" "Was it overconfident before failing?" "Did its confidence match reality?" | Confidence-quality signal for verify/replan routing and overconfidence-aware training updates |

The method can be read as three layers:

- **signal layer:** entropy, rollout agreement, and optional verifier partial score define the confidence signal;
- **policy layer:** low confidence can trigger continue / verify / replan decisions at inference time;
- **training layer:** high-confidence failures are penalized more strongly than low-confidence exploratory failures.

**Why is this not just distillation?** The core method keeps the confidence signal on-policy and grounded by grouped rollouts plus verifier outcomes. An external frontier model can appear only as an ablation, not as the main teacher.

**Why not only use verifier reward?** Verifier reward tells us whether a rollout was right, but not whether the model should have known it was at risk. The proposal uses confidence to improve credit assignment and routing, not to replace the verifier.

**How is confidence calculated in practice?** V1 uses normalized entropy, rollout agreement, and optional verifier partial scores. V2 may add a trainable head, but only after grounded targets are established.

**What models are used for baseline and treatment?** The baseline and treatment use the same open policy family, ideally a Qwen-class model in the 3B–7B range for pilot runs. The difference is the reward shaping and inference policy, not the base model.

**What does the policy update look like?** The policy is updated using grouped RLVR with calibration-aware shaped reward and KL regularization to a reference policy, as defined in Section 3.

**Can an external model rate the trainee model's confidence?** It can, but that is not the preferred method. An external model does not have privileged access to the trainee model's internal uncertainty, and using it in the main loop weakens novelty and moves the setup toward distillation.

**How do we prevent catastrophic forgetting?** We do not assume it away. We measure it with anchor tasks, replay, KL regularization, and preferably adapter-based training for the pilot.

**How does this help long-horizon agents specifically?** It gives an earlier signal that the trajectory is drifting, makes verify/replan more targeted, and punishes confident errors more strongly than ordinary failed rollouts.

**How does this help sample efficiency?** Binary verifier reward discards structure in many failed episodes. Calibration-aware shaping reuses that structure, especially when grouped rollouts expose stable disagreement or stable but wrong agreement.

## References


[1] Shumin Deng, Ningyu Zhang, Nay Oo, and Bryan Hooi. Towards a unified view of answer calibration for multi-step reasoning. *arXiv preprint arXiv:2311.09101*, November 2023. URL https://arxiv.org/abs/2311.09101.

[2] Young-Jin Park, Kristjan Greenewald, Kaveh Alim, Hao Wang, and Navid Azizan. Know what you don't know: Uncertainty calibration of process reward models. *arXiv preprint arXiv:2506.09338*, June 2025. URL https://arxiv.org/abs/2506.09338.

[3] Aman Sharma and Paras Chopra. Think just enough: Sequence-level entropy as a confidence signal for LLM reasoning. *arXiv preprint arXiv:2510.08146*, October 2025. URL https://arxiv.org/abs/2510.08146.

[4] Paul Stangel, David Bani-Harouni, Chantal Pellegrini, Ege Ozsoy, Kamilia Zaripova, Matthias Keicher, and Nassir Navab. Rewarding doubt: A reinforcement learning approach to calibrated confidence expression of large language models. *arXiv preprint arXiv:2503.02623*, March 2025. URL https://arxiv.org/abs/2503.02623.

[5] Jiawei Wang, Jiacai Liu, Yuqian Fu, Yingru Li, Xintao Wang, Yuan Lin, Yu Yue, Lin Zhang, Yang Wang, and Ke Wang. Harnessing uncertainty: Entropy-modulated policy gradients for long-horizon LLM agents. *arXiv preprint arXiv:2509.09265*, September 2025. URL https://arxiv.org/abs/2509.09265.

[6] Tianyang Xu, Shujin Wu, Shizhe Diao, Xiaoze Liu, Xingyao Wang, Yangyi Chen, and Jing Gao. SaySelf: Teaching LLMs to express confidence with self-reflective rationales. *arXiv preprint arXiv:2405.20974*, May 2024. URL https://arxiv.org/abs/2405.20974.

[7] Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, and Zhipeng Wang. Overconfident errors need stronger correction: Asymmetric confidence penalties for reinforcement learning. *arXiv preprint arXiv:2602.21420*, February 2026. URL https://arxiv.org/abs/2602.21420.

[8] Yuanzhao Zhai, Han Zhang, Yu Lei, Yue Yu, Kele Xu, Dawei Feng, Bo Ding, and Huaimin Wang. Uncertainty-penalized reinforcement learning from human feedback with diverse reward LoRA ensembles. *arXiv preprint arXiv:2401.00243*, December 2023. URL https://arxiv.org/abs/2401.00243.

[9] Bonan Zhang, Zhongqi Chen, Bowen Song, Qinya Li, Fan Wu, and Guihai Chen. ConfClip: Confidence-weighted and clipped reward for reinforcement learning in LLMs. *arXiv preprint arXiv:2509.17730*, September 2025. URL https://arxiv.org/abs/2509.17730.

[10] Dengjia Zhang, Xiaoou Liu, Lu Cheng, Yaqing Wang, Kenton Murray, and Hua Wei. SELAUR: Self evolving LLM agent via uncertainty-aware rewards. *arXiv preprint arXiv:2602.21158*, February 2026. URL https://arxiv.org/abs/2602.21158.

[11] Jiaxin Zhang, Prafulla Kumar Choubey, Kung-Hsiang Huang, Caiming Xiong, and Chien-Sheng Wu. Agentic uncertainty quantification. *arXiv preprint arXiv:2601.15703*, January 2026. URL https://arxiv.org/abs/2601.15703.