Self-Critic Policy Optimization (SCPO)

Toward Robust Reward Signals for RL Post-Training of LLMs

Ro

April 2026

Setting the Stage

Domain and Scope

We work in **RL post-training for LLMs** on tasks with a verifier (math, code).

Goal: Develop a unified, robust objective for policy optimization that:

- Solves key pitfalls of GRPO
- Provides a richer reward signal than binary verification
- Plugs directly into existing RLVR workflows

Focus benchmarks: GSM8K, MDPP+, LiveCodeBench v6

The GRPO Baseline

GRPO (DeepSeek) eliminated the critic/value function in favor of a verifier.

Strengths: Simple, stable, no reward model overhead.

Weaknesses:

- **Advantage collapse:** when all rollouts in a group score 0 or all score 1, relative advantage $\to 0$ and gradient signal vanishes
- **Sparse reward:** only 1 bit of information per rollout; intermediate reasoning steps receive no supervision
- **No discrimination:** near-misses and hopeless failures are treated identically
- **Wasted feedback:** all environment information beyond pass/fail is discarded

Self-Distillation: A Partial Fix

Recent work (OPSD, SDPO, SDFT; Jan 2026) uses a **self-teacher pass** with privileged context (verifier result, execution trace) to extract token-level supervision without an external PRM.

Core limitations:

- **Teacher imitation via KL:** forces the student to match teacher outputs. What if the teacher is wrong? Reduces epistemic uncertainty and may suppress emergent reasoning.
- **Mode collapse:** teacher locks onto a subset of solution modes, limiting exploration. Partly driven by reverse KL.
- **Off-policy drift** (debatable): the teacher distribution may diverge from the on-policy student.

Process Reward Models: Another Angle

PRMs reward intermediate steps but introduce their own issues:

- Require a **separate model** (overhead, complexity)
- **Off-policy drift:** static PRM becomes exploitable as the policy evolves
- **Step labels:** need human annotation, which is expensive and brittle

Can we get the benefits of PRMs and self-distillation without inheriting their failure modes?

Desiderata

What We Want

Reward Signal

- Online, on-policy (no drift)
- Richer than binary (process-aware)
- Uses verifier result as grounding
- Evaluative, not generative

Training Dynamics

- Preserves/encourages reasoning
- Mode diversity (pass@k $\uparrow$, not just pass@1)
- Scales with model size
- Scales with compute

Key insight: Frame the self-teacher as an evaluator, not a generator.

Like a student taking a test, getting graded, and retrying—rather than copying the teacher's solution.

Core Research Questions

Questions to Investigate

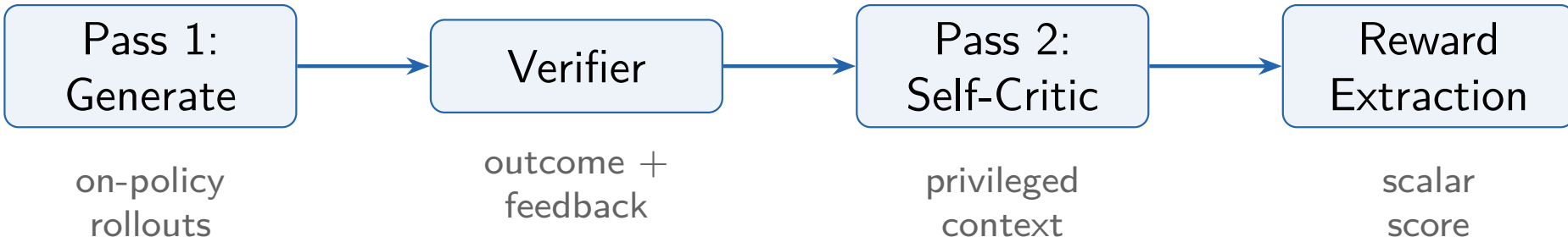
1. **Level of supervision:** Dense/token-level? Sparse/outcome-level? Somewhere in between?
2. **External parameters:** Should we introduce a separate critic, add parameters to the policy (e.g., reward head), or estimate value purely from existing forward-pass quantities?
3. **Reasoning preservation:** Does the self-teacher mechanism degrade or enhance emergent reasoning?
4. **Scaling:** Does performance improve with model size? With additional compute at the self-critic stage?

Potential emergent phenomena:

- Reward hacking detection/mitigation
- Extended reasoning in the self-teacher pass
- Self-teacher improving over the course of training

The SCPO Framework

Two-Pass RL: Overview



Pass 1 generates rollouts normally. The **verifier** produces outcome + feedback. **Pass 2** re-reads the rollout with privileged information and produces a richer reward signal. The policy is updated using this signal within the GRPO objective.

Variant 1: Reward Head on the Policy

Add a small **learned head** to the policy model.

- During the self-teacher pass (with privileged context), the head outputs a scalar reward
- This reward is plugged into the GRPO objective directly
- Acts as a **self-critic** or on-policy value function
- Trained end-to-end: no separate model, stays on-policy by construction

Pros: Minimal parameter overhead; naturally on-policy; single model.

Risks: Head capacity may limit reward expressiveness; training stability.

Variant 2: Estimate Critic from Self-Teacher

No additional parameters. Extract reward from the self-teacher forward pass directly.

- Use divergence between teacher and student distributions (KL, JS)
- Aggregate logit differences at token or span level
- Learn a lightweight operator over intermediate representations
- Potentially use DPO-style preference ranking between rollouts

Pros: Zero additional parameters; computationally lightweight.

Risks: Reward signal may be noisy or poorly calibrated.

Variant 3: LLM-as-a-Judge in Self-Teacher Pass

Prompt the self-teacher to output a reward **in natural language**.

- Self-teacher receives: rollout + verifier outcome + execution trace
- Prompted to evaluate solution quality, identify errors, assign a score
- Natural language critique is parsed into a scalar reward
- Leverages the model's own reasoning capability for evaluation

Pros: Rich, interpretable feedback; uses existing model capabilities.

Risks: Generation overhead; self-evaluation bias; parsing fragility.

Variant 4: Online Critic / Reward Model

Introduce a **separate critic**, but train it online alongside the policy.

- Critic is updated on the fly using rollouts and verifier outcomes
- Avoids the off-policy drift of static PRMs
- Could share a backbone with the policy (asymmetric actor-critic)
- More expressive than a reward head, but adds model overhead

Pros: High reward expressiveness; on-policy by design.

Risks: Training instability (the classic problem GRPO tried to avoid).

Variant 5: Inverse RL / Implicit Reward

Frame reward discovery as an **inverse RL problem**.

Approach A: Treat verifier-passing rollouts as expert demonstrations.

Approach B: Treat self-teacher rollouts as expert demonstrations.

1. Policy generates rollouts for a problem
2. Self-teacher re-runs with privileged context $\rightarrow$ teacher rollouts
3. Train a discriminator (GAIL/AIRL) to distinguish teacher from student
4. Use the discriminator's output as a learned reward
5. Optimize the student policy via GRPO with this reward

Inspiration: RARO (GAIL for non-verifiable domains). We adapt this for the verifiable setting.

Comparison of Variants

Trade-offs at a Glance

Variant	**Extra Params**	**On-Policy**	**Expressiveness**
Reward Head	Minimal	✓	Medium
Critic Estimate	None	✓	Low–Med
LLM-as-Judge	None	✓	High
Online Critic	Full model	✓	High
Inverse RL	Discriminator	✓	High

All variants share the **two-pass structure** and plug into GRPO. The key axis of variation is how the reward is extracted from the self-critic pass.

Related Work

Key References

RL Post-Training

- GRPO (DeepSeek)
- PPO, DPO
- Actor-Critic, A2C/A3C

Self-Distillation

- RLSD, OPSD
- SDPO, SDFT
- HDPO

Reward Modeling

- Process Reward Models
- RARO (GAIL / IRL)

Analysis

- Thinking Machines blog
- On-policy distillation survey (Apr 2026)
- "Does self-distillation degrade reasoning?"

Next Steps

Proposed Plan

1. **Prototype:** Implement the two-pass framework with the reward head variant (simplest)
2. **Baseline:** Compare against vanilla GRPO and self-distillation (OPSD/SDPO) on GSM8K
3. **Iterate:** Test critic-estimate and LLM-as-judge variants
4. **Evaluate:** Track pass@1, pass@k (diversity), reward hacking metrics, reasoning chain quality
5. **Scale:** Evaluate across model sizes; measure compute scaling behavior

Open directions:

- Extension to non-verifiable domains
- Long-horizon RL with extended chains of thought
- Compute-optimal self-critic strategies

Thank You

Self-Critic Policy Optimization

From verifier to robust reward signal:

evaluative self-teaching for RL post-training

Questions & Discussion