SEVA: Self-Evolving Verification Agent with Process Reward for Fact Attribution

Aojie Yuan [1] **Yi Nian** [1] **Haiyue Zhang** [1] **Zijian Su** [2] **Yue Zhao** [1]

Abstract

Hallucination is the reliability bottleneck for LLM-based agents, and fact attribution verifiers are the last line of defense — yet today's verifiers emit only opaque binary labels, leaving agents unable to self-correct and operators unable to audit. We present SEVA, a structured verification agent that emits evidence alignments, step-by-step reasoning chains, calibrated confidence, and a six-category error diagnosis with actionable fixes. Training such an agent with RL is non-trivial: standard binary reward on multi-component output triggers *advantage collapse* — within-group reward variance vanishes and the GRPO gradient disappears. We resolve this with a process reward that decomposes verification quality into five independent components weighted $70/30$ toward process signals, restoring the gradient and inducing an implicit curriculum — the agent first masters verification *behavior* (alignment $0.917 \rightarrow 0.997$, format $72\% \rightarrow 100\%$), then *outcomes* (F1 $64.9 \rightarrow 69.0$). Structured output further enables a Verify→Reflect→Probe→Refine self-evolution loop, which over four rounds on a 7B model surfaces an unexpected structural finding: each round produces a *benchmark-specialist*, not a generalist ($+15$ pp on HaluEval, -10 to -14 pp on TruthfulQA in the same model, persistent at $4\times$ data). On ClearFacts, SEVA-3B matches GPT-4o-mini (69.0 vs. 69.8 F1) while producing substantially richer, auditable output — confirming a principle that should generalize: for any RL task with multi-component generation, *reward granularity must match output granularity*.

[1]University of Southern California, Los Angeles, USA [2]University of Michigan, Ann Arbor, USA. Correspondence to: Aojie Yuan <aojieyua@usc.edu>, Yi Nian <yinian@usc.edu>, Haiyue Zhang <haiyuez@usc.edu>, Zijian Su <simoon@umich.edu>, Yue Zhao <yue.z@usc.edu>.

Accepted at the ICML 2026 Workshop on Trustworthy AI for Good (AI4GOOD), Seoul, South Korea. Copyright 2026 by the author(s). Code: https://github.com/Justin0504/Verifiable_agent

1. Introduction

Despite rapid progress in LLM capabilities, hallucination remains a fundamental barrier to deploying agents in high-stakes domains such as finance, law, and healthcare (Min et al., 2023). Fact attribution verifiers — models that judge whether each claim in an agent's output is supported by its source documents — have emerged as a critical safety layer (Gao et al., 2023; Tian et al., 2024). Systems like MiniCheck (Tang et al., 2024) and ClearCheck (Seo et al., 2025) achieve strong accuracy on this task, but they share a fundamental limitation: they produce only a binary label.

This opacity creates two problems for agents in the wild. First, when a verifier flags a claim, the agent has no basis for *self-correction* — it knows something is wrong, but not whether a percentage was inflated, an entity swapped, or a qualifier fabricated. Second, no human operator can meaningfully *audit* the decision, because the reasoning behind the label is invisible. In safety-critical deployments, an uninterpretable verifier undermines the very trust it is meant to provide.

We introduce SEVA (**S**elf-**E**volving **V**erification **A**gent), which addresses both problems by producing *structured* verification output: evidence alignment spans that ground every judgment in specific text, reasoning chains that trace the logic step by step, and error diagnoses from a six-category taxonomy with actionable fix suggestions. This structured output serves a dual purpose: it makes verification auditable for deployment, and it provides a diagnostic interface for training.

How should such an agent be trained? SFT on teacher-annotated data provides a reasonable starting point, but reinforcement learning — which has driven substantial gains for mathematical reasoning (Shao et al., 2024; Zha et al., 2025) and hallucination reduction (Li et al., 2026) — does not straightforwardly transfer. Applying GRPO (Shao et al., 2024) with binary reward (1 if the label matches, 0 otherwise) to our structured verifier, we find that training stalls entirely: the policy makes no progress beyond SFT across all 350 steps. The culprit is that binary reward compresses all verification quality into a single bit. A response with correct reasoning but the wrong label receives the same

score — zero — as one that produces unparseable garbage. In a GRPO group of $G{=}8$ responses, most therefore score 0, advantage spread contracts to ± 0.05, and the gradient vanishes.

Our response is a **process reward function** $R : \mathcal{V} \times \mathcal{Y} \to [-0.10, 1.28]$ mapping each structured response $\mathbf{v}$ (against gold y^*) to a continuous score over five independent components plus a calibration term:

$$R = \underbrace{w_f R_f + w_a R_a + w_c R_c}_{\text{process (70\%)}} + \underbrace{w_l R_l + w_d R_d}_{\text{outcome (30\%)}} + R_{\text{cal}} \qquad (1)$$

with weights $w_f{=}0.10$, $w_a{=}w_c{=}0.30$, $w_l{=}w_d{=}0.15$, and an asymmetric calibration $R_{\text{cal}} = +\hat{\gamma}{\cdot}0.15$ if $\hat{y}{=}y^*$ else $-\hat{\gamma}{\cdot}0.10$, which rewards calibrated correctness more than it penalizes calibrated error. Each $R_x \in [0, 1]$ is computed independently from a different region of $\mathbf{v}$ (App. C), so the components are weakly correlated; this independence is what creates the smooth four-level reward landscape (Fig. 6, Tab. 1) that resolves the collapse. A response with sound reasoning but the wrong label scores ~ 0.63 rather than 0.0, restoring $\sigma > 0$ in Eq. 2.

The results confirm that process reward unlocks what binary reward cannot. GRPO lifts alignment quality to 0.997, format compliance to 100%, and F1 to 69.0 (+4.1 over SFT). An implicit curriculum emerges in the training dynamics: the agent masters verification *behavior* within 150 steps, then spends the remaining 200 steps refining verification *outcomes* — without any explicit scheduling.

Structured output confers a further advantage: it makes the agent's failures *transparent*. When SEVA misclassifies a claim, its evidence alignments and error diagnoses pinpoint which error category was missed and where grounding broke down. We channel this diagnostic signal into a **Verify→Reflect→Probe→Refine** self-evolution loop that generates targeted adversarial data for the agent's weakest error categories, and iterate it four times on the 7B model. A surprising empirical finding emerges: each round yields a *benchmark-specialist* rather than a strictly better generalist, and the asymmetric trade-off persists at $4\times$ training-data scale (7,787 samples in Round 4) — confirming that the effect is data-distribution-induced rather than overfitting. This finding is itself only visible because the structured output exposes per-category error dynamics that aggregate accuracy would hide, and it sits uneasily with the monotonic-improvement assumption implicit in Self-Refine / STaR-style self-evolution literature.

Our contributions are threefold.

1. **SEVA: a structured verifier that's also auditable.** A 3B agent whose output — alignments, reasoning chain, calibrated confidence, six-category error diagnosis with fixes — matches GPT-4o-mini's accuracy (69.0 vs. 69.8 F1) while producing the substrate every downstream operator needs (§2.2, §3).
2. **A process reward that turns RL on structured output from impossible to possible.** We formalize *advantage collapse* (Prop. 1) as the failure mode of binary reward on multi-component generation, then resolve it with a five-component decomposition (Prop. 2); the resulting reward landscape yields an *implicit curriculum* — behavior before outcomes — without any explicit scheduling (§2.3, §3.5).
3. **A self-evolution loop that reveals a structural property of iterative refinement.** Verify→Reflect→Probe→Refine, iterated four rounds on a 7B model, surfaces a finding that contradicts the monotone-improvement assumption of Self-Refine / STaR: each round produces a *benchmark-specialist*, not a generalist — $+15$ pp HaluEval, -10 to -14 pp TruthfulQA in the *same* model, robust at $4\times$ training data, visible only because structured output exposes per-category dynamics (§2.5).

2. Method

2.1. Problem Formulation and Overview

Given a claim c and source document d, a fact attribution verifier produces a judgment about whether c is supported by d. Existing verifiers output a single binary label (Tang et al., 2024; Seo et al., 2025). We instead require the agent to produce structured output $\mathbf{v} = (A, C, y, \gamma, e, s)$ that makes verification auditable and failures diagnosable (Figure 1).

Building such an agent via SFT is straightforward, but pushing it further with RL is not. We show that GRPO with binary reward fails entirely on this output format (§2.3), and design a process reward that resolves this failure (§2.3). The structured output further enables a self-evolution loop for iterative improvement (§2.5).

2.2. Structured Verification Schema

The output $\mathbf{v}$ comprises four complementary components:

Evidence alignment A: a list of $(c_i, d_i, \text{status}_i)$ triples mapping claim spans to source spans, with status $\in$ {match, mismatch, not_found}. Each entry forces the agent to anchor its judgment in specific text rather than forming a holistic impression.

Reasoning chain C: step-by-step verification where each step examines a claim part against source evidence, producing a judgment $\in$ {supported, not_supported, partially_supported} and a natural language explanation.

Label and confidence: a binary label y paired with calibrated confidence $\gamma \in [0, 1]$.

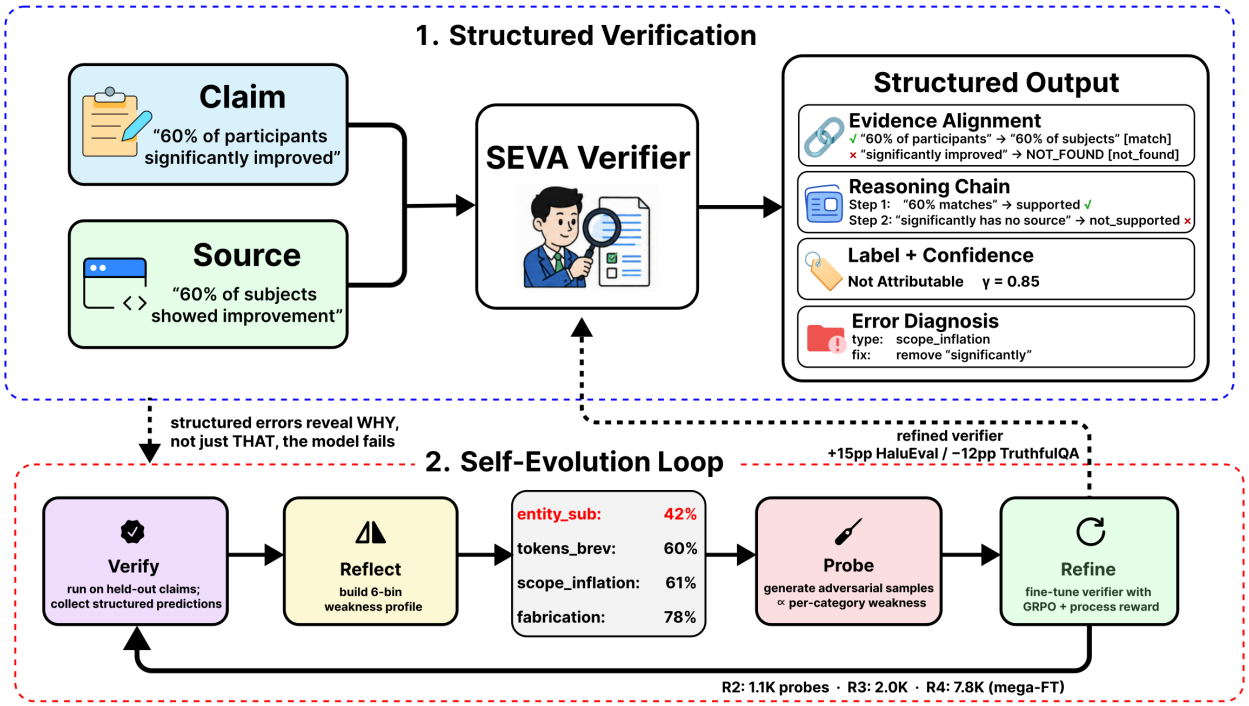

Figure 1. **SEVA overview.** *Top:* Given a claim-source pair, the verifier produces structured output — evidence alignments, reasoning chains, calibrated confidence, and error diagnosis. *Bottom:* Self-evolution loop. Structured errors reveal *why* the model fails (not just *that* it fails), enabling targeted adversarial data generation focused on the weakest error types.

Error diagnosis: when y=Not Attributable, an error type e drawn from a six-category taxonomy (numerical exaggeration, negation flip, scope inflation, temporal shift, entity substitution, fabrication) together with a fix suggestion s.

This schema serves two purposes relevant to agents in the wild. First, it makes verification *auditable*: a human operator can inspect alignments and reasoning to judge whether the verdict is trustworthy. Second, it makes failures *diagnosable*: when the agent errs, the structured output pinpoints which evidence was mishandled, feeding the self-improvement loop in §2.5.

2.3. From Binary Reward Failure to Process Reward

The failure of binary reward. Binary reward assigns 1.0 when the predicted label matches the gold label and 0.0 otherwise. For structured output, this produces a degenerate training signal. In a GRPO group of G=8 responses: (1) 28% fail JSON parsing — the SFT model produces valid JSON only 72% of the time, and all of these score 0; (2) among valid responses, ∼35% predict the wrong label, also scoring 0; (3) in a typical group, 5–7 of 8 responses receive zero reward. GRPO computes advantages relative to the group mean. For a group of G responses with rewards $\{r_1, \ldots, r_G\}$, the normalized advantage of response i is:

$$\hat{A}_i = \frac{r_i - \mu}{\sigma + \epsilon}, \quad \mu = \frac{1}{G}\sum_j r_j, \quad \sigma = \mathrm{std}(\{r_j\}) \tag{2}$$

When binary reward produces $r_j \in \{0, 1\}$ with most $r_j = 0$, both μ and σ are near zero, and $\hat{A}_i \approx 0$ for all i — the policy gradient $\nabla_\theta J \propto \sum_i \hat{A}_i \nabla_\theta \log \pi_\theta$ vanishes regardless of model parameters.

This failure is *structural*, not incidental. Increasing group size does not help — the problem is near-uniform scores, not insufficient sampling. And the failure is not specific to verification: any RL task whose output has multiple required components will exhibit advantage collapse under binary reward whenever the model cannot reliably produce all components simultaneously. We formalize the mechanism below; a proof sketch is given in Appendix C.7.

Proposition 1 (Binary-Reward Advantage Collapse)**.** *Let $r_1, \ldots, r_G \in \{0, 1\}$ be i.i.d. Bernoulli rewards in a GRPO group of size G with success probability $q = \Pr[r_j{=}1]$. The expected within-group variance is*

$$\mathbb{E}[\sigma^2] = \tfrac{G}{G-1}\, q\,(1-q), \tag{3}$$

and as $q \to 0^+$ or $q \to 1^-$, $\sigma \xrightarrow{a.s.} 0$, hence

$$\hat{A}_i = \frac{r_i - \mu}{\sigma + \epsilon} \xrightarrow{a.s.} 0 \qquad \forall\, i, \tag{4}$$

Table 1. Reward landscape. Process reward produces four distinct quality levels where binary reward sees only two, enabling fine-grained advantage estimation within each GRPO group.

Response quality	Process	Binary
Correct label + good reasoning	∼1.13	1.0
Good reasoning, wrong label	∼0.63	0.0
Correct label, poor reasoning	∼0.28	1.0
Unparseable output	0.0	0.0

and the expected policy gradient

$$\nabla_\theta J \;=\; \mathbb{E}\left[\sum_{i=1}^{G} \hat{A}_i \, \nabla_\theta \log \pi_\theta(\mathbf{v}_i)\right] \;\to\; 0 \tag{5}$$

regardless of *model parameters θ or group size G.*

Proposition 2 (Process-Reward Variance Lower Bound)**.** *Let $R = \sum_{k=1}^{K} w_k R_k$ be the aggregate process reward with components $R_k \in [0,1]$ and positive weights $w_k > 0$. By the variance identity for linear combinations,*

$$\sigma^2(R) \;=\; \sum_{k=1}^{K} w_k^2 \, \sigma_k^2 \;+\; 2 \sum_{k<\ell} w_k \, w_\ell \, \mathrm{Cov}(R_k, R_\ell). \tag{6}$$

Unless the components are perfectly anti-correlated, the cross-terms cannot drive $\sigma^2(R)$ to zero unless every σ_k=0. In particular, if any single component k^* has $\sigma_{k^*}^2 > 0$ and is uncorrelated with the rest,

$$\sigma^2(R) \;\geq\; w_{k^*}^2 \, \sigma_{k^*}^2 \;>\; 0, \tag{7}$$

and the GRPO gradient is non-vanishing.

Why these matter. Prop. 1 pinpoints *the* failure mode of binary reward on structured output, and Prop. 2 guarantees that process reward escapes it by construction. Empirically, $q \approx 0.37$ at SFT-init gives $\mathbb{E}[\sigma^2] \lesssim 0.27$ (Eq. 3) and shrinks to $\sigma \approx 0.05$ by step 350 (Tab. 18) — the gradient effectively dies under binary reward. Under process reward, format errors at $\sim 28\%$ keep $\sigma_f > 0$ at every step we observe, so Eq. 7 delivers $\sigma^2(R) > 0$ throughout training; the smooth four-level landscape of Tab. 1 is the geometric consequence. The argument is task-agnostic: any RL setting whose output has K required components inherits the same dichotomy, so a process-style decomposition is the structural fix wherever it applies.

The reward landscape (Tab. 1) inverts the binary ranking: "good reasoning, wrong label" scores 0.63 (vs. 0) and "correct label, poor reasoning" only 0.28 (vs. 1) — binary reward effectively pays for lucky guesses; process reward pays for genuine verification work.

Why 70/30, and how each component scores. The split forces the agent to do substantive verification *before* the label becomes the easy lever: with outcome dominating, the model would learn to guess labels and produce incoherent reasoning around them. Each R_x scores independently — R_f on JSON validity, R_a on per-span grounding, R_c on per-step judgment and citation, R_l on label match, R_d on error type and fix; an asymmetric calibration term $R_{\text{cal}} \in \{+\gamma{\cdot}0.15, -\gamma{\cdot}0.10\}$ rewards confident correctness and penalizes overconfident error. Algorithm 1 gives the full computation; per-component rubrics are in Appendix C.

Algorithm 1 Process Reward Computation

Require: Response r, gold label y^*
Ensure: Reward $R \in [-0.10, 1.28]$
Parse r as JSON $\to \hat{v}$
if parse fails **then**
  return $R = 0.0$
end if
$R_f \leftarrow$ ScoreFormat($\hat{v}$) {0 / 0.2 / 0.5 / 1.0}
$R_a \leftarrow \frac{1}{|A|} \sum_{a \in A}$ ScoreAlign(a)
$R_c \leftarrow \frac{1}{|C|} \sum_{s \in C}$ ScoreStep(s) + LenBonus
$R_l \leftarrow \mathbf{1}[\text{normalize}(\hat{y}) = y^*]$
$R_d \leftarrow$ ScoreDiagnosis($\hat{e}, \hat{s}, y^*$)
$R \leftarrow 0.10\, R_f + 0.30\, R_a + 0.30\, R_c + 0.15\, R_l + 0.15\, R_d$
if $\hat{y} = y^*$ **then**
  $R \leftarrow R + \hat{\gamma} \times 0.15$ {calibration bonus}
else
  $R \leftarrow R - \hat{\gamma} \times 0.10$ {overconfidence penalty}
end if
return R

2.4. Training Pipeline

SEVA is trained in two phases. **SFT:** GPT-4o-mini annotates 4,992 ANLI examples with structured output (92% format-valid); Qwen2.5-3B-Instruct (Qwen Team, 2025) is fine-tuned for 3 epochs at lr $= 2{\times}10^{-5}$. **GRPO:** the SFT checkpoint seeds 5 epochs (∼350 steps) of process-reward GRPO with G=8, T=1.2, β=0.001, lr $= 2{\times}10^{-6}$, on veRL (Sheng et al., 2025) with FSDP on 2×RTX 6000 Ada (∼28 GPU-hours total). The low GRPO lr and small β jointly preserve the SFT-established format while leaving room for the policy to explore; we found this balance via a small sweep (β=0.01 over-regularized; β=0 admitted format-gaming). We additionally train a 7B variant via two-stage SFT (binary NLI → structured) with LoRA-128 (Hu et al., 2022); full hyperparameters in Appendix A.

2.5. Self-Evolution via Structured Diagnostics

Structured output exposes which aspect of verification failed when the agent misclassifies; we channel this signal through a **Verify→Reflect→Probe→Refine** loop (Fig. 1, bottom), borrowing the principle of *functional separation* from MARCH (Li et al., 2026) but applying it across loop stages rather than agents. Reflect aggregates error diagnoses into a 6-bin weakness profile; Probe allocates adversarial

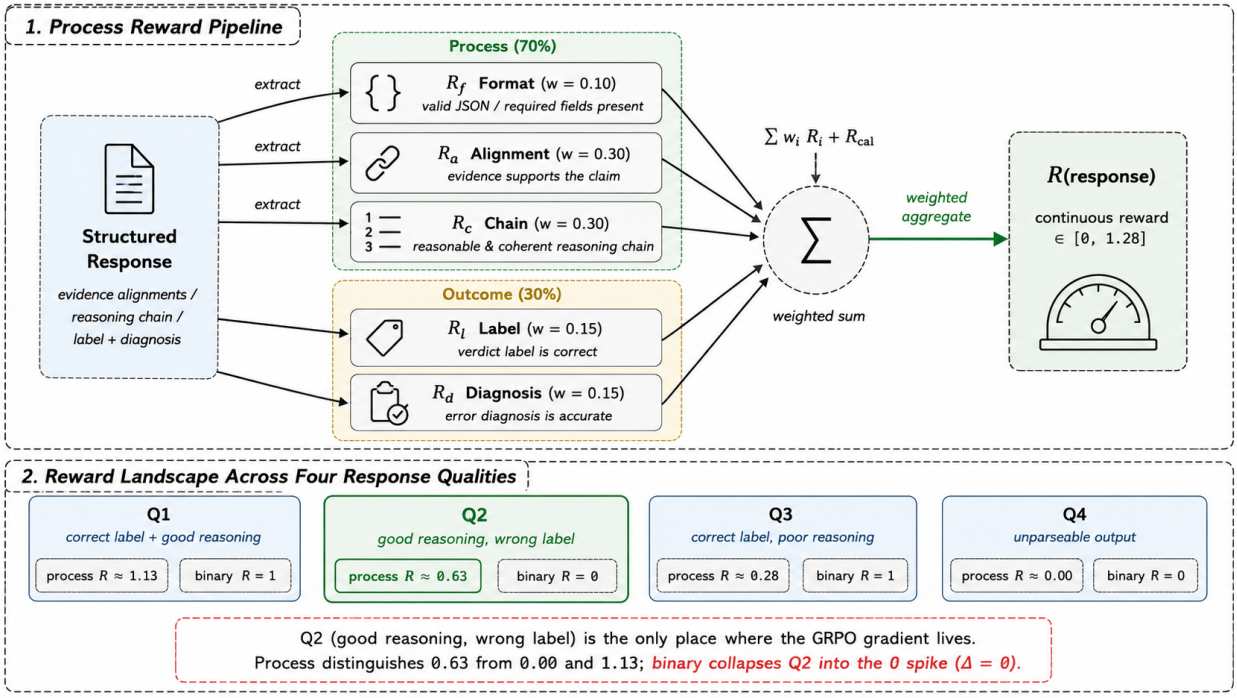

Figure 2. **Process reward scoring.** Each structured output component (left) maps to an independently scored reward term (right). A response with correct reasoning but the wrong label scores 0.63 under process reward vs. 0.0 under binary — this gap is what provides GRPO with meaningful gradients.

generation budget proportional to per-category weakness, giving weak bins $\sim 3\times$ the budget of strong ones (e.g., entity_sub at 42% acc gets $\sim 3\times$ fabrication's at 78%).

GRPO training itself constitutes Round 0: the process reward continuously assesses structural quality, and rollout sampling explores the agent's decision boundary. We then iterate four additional rounds on the 7B Step150 seed: **Round 1** injects extracted verification rules into the prompt (no parameter update); **Round 2** performs LoRA SFT on 1,122 adversarial probes; **Round 3** performs full FT on 2,013 mixed samples (adversarial + replay); **Round 4** ("mega-FT") extends Round 3 with 7,787 mixed samples to test whether more diverse adversarial data closes any remaining gap. Pseudocode for the four-stage loop is in App. L (Algorithm 2).

Specialists, not generalists. We expected four refinement rounds to yield a monotonically improving generalist; we observe a sharper *specialization fingerprint* instead (Table 2). Round 2 lifts HaluEval by $+10.9$ pp but drops TruthfulQA by -10.2; Round 3 sharpens to $+14.3/-13.8$; Round 4 holds at $+14.9/-12.4$ despite $4\times$ more data. Persistence at $4\times$ scale rules out trivial overfitting and identifies the effect as *data-distribution-induced*: probes drawn from a ClearFacts-style weakness profile push the model toward those failure modes and away from TruthfulQA's distribution. *When probes come from a single source distribution, specialization is the dominant mode of iterative refinement.*

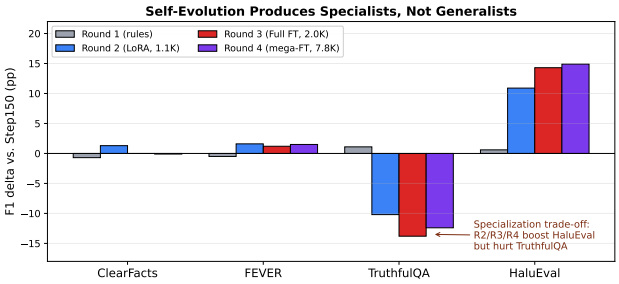

Figure 3. **Self-evolution produces specialists, not generalists.** F1 deltas vs. the Step150 GRPO seed across four benchmarks and four refinement rounds. The asymmetric trade-off on TruthfulQA vs. HaluEval (Rounds 2–4) sharpens with each round and persists at $4\times$ training-data scale (Round 4, 7,787 samples), confirming the effect is data-distribution-induced rather than overfitting. Absolute F1 numbers in Appendix L, Table 22.

Does the loop actually work? Three properties make the data-volume reading hard to sustain: *(i)* the trade-off is monotone in both directions ($+10.9 \rightarrow +14.9$ HaluEval, $-10.2 \rightarrow -12.4$ TruthfulQA — a non-functional loop would sign-flip); *(ii)* per-benchmark gains track Probe-stage budget allocation, not raw sample count; *(iii)* Round 4 has $7\times$ Round 2's data but adds only $+4$ pp on HaluEval, exhibiting saturation rather than the unbounded growth of data-volume

Table 2. Self-evolution produces *specialists*, not generalists. Each row shows macro-F1 deltas vs. Step150 across four benchmarks. The opposite-sign gains on TruthfulQA vs. HaluEval (rows 2–4) persist at $4\times$ training-data scale (Round 4), confirming the effect is data-distribution-induced.

	ClearF.	FEVER	TrQA	HaluE.
Round 1 (rules)	-0.7	-0.5	$+1.1$	$+0.6$
Round 2 (LoRA)	$+1.3$	$+1.6$	$\mathbf{-10.2}$	$+10.9$
Round 3 (FT)	0.0	$+1.2$	$\mathbf{-13.8}$	$+14.3$
Round 4 (mega-FT)	-0.1	$+1.5$	$\mathbf{-12.4}$	$+14.9$

overfitting. The clean isolation ablation — swapping the weakness-guided Probe for a same-budget random sampler — is out of scope for this submission (§4.3); the loop's response to its signal is consistent with a working mechanism (full analysis in App. L). This specialization fingerprint is itself only visible because structured output exposes per-category dynamics; the same asymmetry would be invisible in aggregate accuracy, motivating downstream architectural responses (e.g., per-domain routing across rounds) we explore in follow-up work.

3. Experiments

Having laid out the architecture (§2.2), the reward (§2.3), and the self-evolution loop (§2.5), we now stress-test SEVA on four axes that any deployed verifier must clear: **accuracy** against established binary baselines, **generalization** across benchmarks with different failure modes, **structural reliability** of the produced output, and **training dynamics** under both reward designs.

3.1. Setup

We evaluate on ClearFacts (Seo et al., 2025) (1,590 samples; our primary metric), FEVER (Thorne et al., 2018) (200), TruthfulQA (Lin et al., 2022) (400), and HaluEval (Li et al., 2023) (200). Together these cover claim-source attribution, encyclopedic verification, common misconceptions, and LLM-generated hallucinations — four distinct distributions chosen to probe whether SEVA's structural advantages survive across error types.

Baselines include binary verifiers reported by Seo et al. (2025): MiniCheck-7B (81.2 F1), ClearCheck-8B ($\sim$84 F1), and Llama-3.1-8B zero-shot (67.2 F1). For structured comparison we evaluate GPT-4o-mini with zero-shot SEVA prompting and MiniCheck-Flan-T5-Large (770M). We report macro F1, accuracy, and structural quality (alignment quality R_a, chain quality R_c, format compliance rate).

3.2. Main Results

Table 3 presents ClearFacts results. With process reward, GRPO lifts SEVA-3B from 64.9 to 69.0 F1 (+4.1), narrowing the gap with GPT-4o-mini (69.8) to under one point. Importantly, SEVA produces substantially richer output — grounded evidence spans, multi-step reasoning, and a six-category error taxonomy — that zero-shot prompting of GPT-4o-mini captures only partially.

Table 3. ClearFacts results. SEVA models produce full structured output; binary baselines from Seo et al. (2025). The gap with MiniCheck reflects training data scale (5K structured vs. 57K binary) rather than a limitation of the approach.

Model	Size	Output	Acc	F1
Binary-label verifiers				
Llama-3.1 (0-shot)	8B	binary	–	67.2
MiniCheck	7B	binary	–	81.2
ClearCheck	8B	binary	–	$\sim$84
Structured verifiers				
GPT-4o-mini (0-shot)	–	struct	69.9	69.8
MiniCheck-Flan-T5	770M	binary	68.3	68.3
Ours				
SEVA-SFT	3B	struct	65.2	64.9
SEVA-GRPO	3B	struct	**69.6**	**69.0**
SEVA-SFT (LoRA-128)	7B	struct	68.6	68.5

Table 4. Macro F1 across four benchmarks. GRPO yields large gains on balanced benchmarks but introduces a negative-prediction bias on skewed ones (§4).

	Out.	ClearF.	FEVER	TrQA	HaluE.
GPT-4o-mini	struct	69.8	**91.0**	48.6	34.0
MiniCheck-FT5	binary	68.3	87.1	59.5	**42.4**
SEVA-SFT (3B)	struct	64.9	76.3	72.1	42.0
SEVA-GRPO (3B)	struct	**69.0**	84.9	**82.7**	39.4

At 7B scale, SFT with LoRA-128 reaches 68.5 F1 without any RL, nearly matching 3B GRPO. Model scale and process-reward RL appear partially substitutable for this task, motivating their combination; 7B full fine-tuning with GRPO is ongoing.

The gap to MiniCheck-7B (81.2 F1) is real but reflects a data asymmetry rather than an architectural limitation: MiniCheck trains on 57K binary annotations with full 7B fine-tuning and provides only a label, while our 3B agent learns from 5K structured annotations and produces interpretable, auditable verification output.

3.3. Generalization Across Benchmarks

GRPO's gains are largest on class-balanced benchmarks ($+8.6$ FEVER, $+10.6$ TruthfulQA). The 34-point TruthfulQA gap over GPT-4o-mini (82.7 vs. 48.6) traces directly to R_c's per-step source-citation requirement: GPT-4o-mini falls back on parametric knowledge when claims "sound right," while SEVA is forced to ground every step in the

Table 5. Structural quality on ClearFacts. After GRPO, alignment and chain quality approach 1.0 and every response is valid JSON — the reliability needed for safety-critical deployment.

	Align	Chain	Format	Δ F1
SEVA-SFT	0.917	0.917	72%	–
SEVA-GRPO	**0.997**	**0.995**	**100%**	+4.1

Claim: "60% of participants significantly improved"
Source: "60% of subjects showed improvement"

Binary verifier: Not Attributable (no explanation)

SEVA-*GRPO structured output:*

Align: "60% of participants" → "60% of subjects" [match]; "significantly improved" → NOT_FOUND

Chain: Step 1: supported (percentage matches); Step 2: not_supported (qualifier absent)

Label: Not Attributable, confidence 0.85

Diag: scope_inflation; fix: remove "significantly"

Figure 4. **Binary vs. structured verification.** The binary verifier is correct but uninformative. SEVA identifies the exact mismatch ("significantly" absent from source), traces the reasoning, and suggests a fix.

document. HaluEval (−2.6 vs. SFT) is the exception — the agent over-predicts "Not Attributable," a reward-induced bias we trace in §4.

3.4. Structural Quality

Process reward drives structural quality to near-perfect levels (alignment 0.997, chain 0.995, format 100% — Tab. 5); a verifier whose output fails to parse 28% of the time under SFT cannot serve as a dependable safety component, so this reliability is itself load-bearing for deployment.

Qualitative gap. Fig. 4 makes the deployment-relevance concrete: on the same input, the binary verifier returns "Not Attributable" with no explanation, while SEVA pinpoints the exact mismatch ("significantly" absent from source), traces step-by-step reasoning, names the error category, and suggests a fix — everything a downstream correction module or human reviewer needs to act, packaged in ∼120 tokens of JSON.

3.5. Training Dynamics and Implicit Curriculum

Binary reward's advantage spread decays from ±0.12 to ±0.04 over 350 steps while its mean barely moves (0.38 → 0.41); process reward sustains spread above ±1.6 throughout, with mean climbing from 1.01 to 1.12 (full trajectory in App. E). A training-time ordering we did not design for

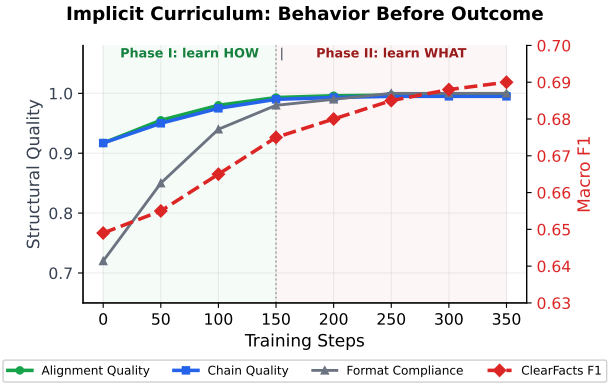

Figure 5. **Implicit curriculum.** Alignment, chain, and format quality saturate by step ∼150 while F1 continues climbing through step 350 — the agent learns *how* to verify before *what* to predict. Reward trajectories and advantage-spread plots in App. E (Fig. 6, Tab. 18).

emerges from this contrast (Fig. 5): alignment and format saturate by step ∼150 (0.917 → 0.997, 72% → 100%) while F1 continues climbing through step 350 (64.9 → 69.0). The agent masters verification *behavior* before *outcomes* — a natural difficulty asymmetry between pattern-level skills and semantic reasoning, amplified by the 70/30 weighting. Mathematical PRMs (Lightman et al., 2024) show a parallel effect on *sequential* steps; we extend the principle to *parallel* components.

4. Ablation and Analysis

The headline numbers establish that SEVA works; we now ask *why*. Two questions matter: is process reward genuinely the load-bearing design choice (vs. just "GRPO with extra steps"), and where does SEVA still fail systematically? The ablation isolates the first, the error analysis surfaces the second — together they motivate the deployment caveats in §4.3.

4.1. Ablation Study

Table 6 isolates the design choice that matters. Replacing process reward with binary reward (every other GRPO setting identical: G=8, T=1.2, β=0.001, 350 steps) yields < 65 F1 and no structural improvement over SFT — 350 steps of policy optimization producing zero gain because the advantage signal is too weak to learn from. The structural metrics are diagnostic: binary GRPO leaves alignment quality at < 0.92 and format compliance at ∼ 72%, exactly the SFT levels, confirming that the policy never updates. Process GRPO simultaneously drives alignment to 0.997 and format to 100% while improving F1 — a tri-directional gain only possible when the gradient is non-degenerate (Prop. 2). Process reward is therefore not an incremental enhance-

Table 6. Ablation on ClearFacts. Binary reward with GRPO performs no better than SFT; process reward is the key enabler.

Configuration	F1	Align	Format
SEVA-GRPO (process reward)	**69.0**	0.997	100%
SEVA-GRPO (binary reward)	<65	<0.92	∼72%
SEVA-SFT (no RL)	64.9	0.917	72%

ment; it is a prerequisite for applying GRPO to structured output, and the three-row contrast in Tab. 6 is the empirical realization of the theoretical dichotomy in Eq. 4–7.

4.2. Reward Asymmetry and Negative-Prediction Bias

GRPO with process reward over-predicts "Not Attributable" (35.9% false positives on ClearFacts; confusion matrix and error-type distribution in App. D). The cause is structural: R_d gives negative predictions a two-part signal (error type + fix) while positive ones collapse to a scalar (1.0 for correct omission), exposing more "reward surface" to negative predictions and biasing the policy under uncertainty. This helps on balanced benchmarks (FEVER, TruthfulQA) and hurts on positively skewed ones (HaluEval); the catch-all `fabrication` diagnosis at 36.7% of negative predictions is its empirical signature. Label-conditional reward normalization is the natural fix; we leave it to future work.

4.3. Limitations

Four caveats bound the claims. *(i)* Two ablations would strengthen the self-evolution evidence and are out of scope: a Probe-distribution control (random vs. weakness-guided) and a cross-distribution probe mix. *(ii)* GRPO is applied only at 3B; the 7B variant uses LoRA, so the scale–RL combination in Table 3 remains untested at full 7B FT. *(iii)* The 70/30 split was chosen on principled grounds (App. I reports a coarse sweep); a finer-grained search and non-uniform training schedules are future work. *(iv)* The negative-prediction bias induced by R_d's asymmetric reward surface (§4) should be addressed before deployment; label-conditional reward normalization is the natural starting point.

5. Related Work

SEVA sits at the intersection of three previously-uncombined lines. **Fact attribution verification** via NLI transfer (Tang et al., 2024), unified alignment (Zha et al., 2023), and refined benchmarks (Seo et al., 2025) is accurate but unstructured and SFT-only; we retain the benchmarks, add structured output + RL. **RL for reasoning** with GRPO (Shao et al., 2024; Zha et al., 2025) and hallucination detection (MARCH (Li et al., 2026), Dr. Zero (Yue et al., 2026)) assumes single-answer output where correctness reward suffices; that assumption breaks for multi-component generation. **Process reward models** for math (Lightman et al., 2024; Wang et al., 2024) score *sequential* step dependencies; we score *parallel* components, an independence that lets us weight and normalize each separately to produce the smooth landscape GRPO needs.

6. Discussion

Scale and process-reward RL are complementary, not redundant: 7B-SFT-LoRA-128 (68.5 F1) and 3B-GRPO (69.0 F1) reach the same accuracy through different routes, and their combination should close the gap to MiniCheck-7B's 81.2 — a hypothesis we are actively testing. Beyond accuracy, the five-component decomposition is itself dual-use: it extracts five gradients per response where binary reward extracts one, and exposes the per-category dynamics under which the specialization fingerprint becomes visible at all. For deployment, an unparseable response is functionally indistinguishable from a wrong one, so the 28%→ <1% format-error drop is as load-bearing as the F1 gain; safety-critical pipelines can early-stop at step 150 (already format-reliable, not yet F1-saturated) and route the structured diagnosis — taxonomy, alignment, fix — to an auditor as a *contract* between upstream generator and downstream judge, the substrate any trustworthy agent pipeline ultimately needs.

7. Conclusion

A 3B SEVA matches GPT-4o-mini on ClearFacts (69.0 vs. 69.8 F1) at 100% format compliance while producing auditable structured output — alignments, reasoning chains, calibrated confidence, six-category error diagnosis with fixes. The enabler is a process reward that resolves binary reward's advantage collapse on multi-component generation (Prop. 1–2); the surprise is a monotone, signed specialization fingerprint under iterative refinement, visible only because per-category dynamics are exposed by structured output. Three principles should transfer wherever agents must explain, justify, and improve under audit — *reward granularity matches output granularity*, *structured output is a dual-use asset*, *iterative self-improvement drifts toward specialization under single-sourced probes* — favoring architectural responses over more training rounds.

References

Gao, L., Dai, Z., Pasupat, P., Chen, A., Chaganty, A. T., Fan, Y., Zhao, V., Lao, N., Lee, H., Juan, D.-C., and Guu, K. RARR: Researching and revising what language models say, using language models. In *Proceedings of ACL*, 2023.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang,

S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *Proceedings of ICLR*, 2022.

Lacoste, A., Luccioni, A., Schmidt, V., and Dandres, T. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019.

Li, J., Cheng, X., Zhao, W. X., Nie, J.-Y., and Wen, J.-R. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of EMNLP*, 2023.

Li, Z., Zhang, Y., Cheng, P., Song, J., Zhou, M., Li, H., Hu, S., Qin, Y., Zhao, E., Jiang, X., and Jiang, G. MARCH: Multi-agent reinforced self-check for LLM hallucination. *arXiv preprint arXiv:2603.24579*, 2026.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. In *Proceedings of ICLR*, 2024.

Lin, S., Hilton, J., and Evans, O. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of ACL*, 2022.

Min, S., Krishna, K., Lyu, X., Lewis, M., Yih, W.-t., Koh, P. W., Iyyer, M., Zettlemoyer, L., and Hajishirzi, H. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of EMNLP*, 2023.

Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., and Kiela, D. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of ACL*, 2020.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

Seo, W., Han, S., Jung, J., Newman, B., Lim, S., Lee, S., Lu, X., Choi, Y., and Yu, Y. Verifying the verifiers: Unveiling pitfalls and potentials in fact verifiers. In *Proceedings of COLM*, 2025. arXiv preprint arXiv:2506.13342.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., and Guo, D. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Sheng, G., Zhang, C., Ye, Z., Wu, X., Zhang, W., Zhang, R., Peng, Y., Lin, H., and Wu, C. HybridFlow: A flexible and efficient RLHF framework. In *Proceedings of EuroSys*, 2025. arXiv preprint arXiv:2409.19256.

Strubell, E., Ganesh, A., and McCallum, A. Energy and policy considerations for deep learning in NLP. In *Proceedings of ACL*, 2019.

Tang, L., Laban, P., and Durrett, G. MiniCheck: Efficient fact-checking of LLMs on grounding documents. In *Proceedings of EMNLP*, 2024. arXiv preprint arXiv:2404.10774.

Thorne, J., Vlachos, A., Christodoulopoulos, C., and Mittal, A. FEVER: A large-scale dataset for fact extraction and VERification. In *Proceedings of NAACL-HLT*, 2018.

Tian, K., Mitchell, E., Yao, H., Manning, C. D., and Finn, C. Fine-tuning language models for factuality. In *Proceedings of ICLR*, 2024.

Wang, P., Li, L., Shao, Z., Xu, R., Dai, D., Li, Y., Chen, D., Wu, Y., and Sui, Z. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of ACL*, 2024.

Yue, Z., Upasani, K., Yang, X., Ge, S., Nie, S., Mao, Y., Liu, Z., and Wang, D. Dr. Zero: Self-evolving search agents without training data. *arXiv preprint arXiv:2601.07055*, 2026.

Zha, K., Gao, Z., Shen, M., Hong, Z.-W., Boning, D. S., and Katabi, D. RL tango: Reinforcing generator and verifier together for language reasoning. In *Proceedings of NeurIPS*, 2025. arXiv preprint arXiv:2505.15034.

Zha, Y., Yang, Y., Li, R., and Hu, Z. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of ACL*, 2023.

A. Implementation Details

A.1. Hardware and Compute

Experiments were conducted on a local server with 2×NVIDIA RTX 6000 Ada (48 GB each), plus 1×A100 80G for 7B variants and self-evolution rounds. Table 7 summarizes the compute budget for the full pipeline (3B SFT+GRPO, 7B SFT, and the four self-evolution rounds reported in §2.5); the carbon footprint estimate is in Appendix N.

Table 7. Compute budget for the full pipeline. The 3B-only subset (∼28 GPU-hr) is reproducible on a single multi-GPU workstation; the full pipeline including four self-evolution rounds on 7B requires ∼130 GPU-hr.

Experiment	GPUs	Time	Hrs
3B SFT (full FT)	2×Ada	2h	4
3B GRPO (350 steps)	2×Ada	8h	16
7B SFT (LoRA-64)	1×A100 80G	3h	3
7B SFT (LoRA-128)	1×A100 80G	4h	4
Self-evolution rounds (7B, §2.5)			
SE Round 1 (rules, no update)	–	–	0
SE Round 2 (LoRA-64, 1.1K)	1×A100 80G	5h	5
SE Round 3 (Full FT, 2.0K)	1×A100 80G	21h	21
SE Round 4 (mega-FT, 7.8K)	1×A100 80G	72h	72
Eval (per benchmark)	1×Ada	20m	0.3
Adversarial probe generation	GPT-4o-mini API	–	–
3B-only subtotal			∼28
Full pipeline total			∼130

A.2. SFT Hyperparameters

Table 8. SFT hyperparameters.

Parameter	3B (full)	7B (LoRA)
Base model	Qwen2.5-3B	Qwen2.5-7B
Epochs	3	3
Batch (per GPU)	4	4
Grad. accum.	4	4
Eff. batch	16	16
Learning rate	2e-5	2e-5 / 5e-5
Scheduler	Cosine	Cosine
Warmup ratio	0.05	0.05
Weight decay	0.01	0.01
Max seq. len	1280	1280
Precision	bf16	bf16
Grad. ckpt.	✓	✓
LoRA-specific (7B only)		
LoRA rank	–	64 / 128
LoRA alpha	–	128
LoRA dropout	–	0.05
Target mods	–	q,k,v,o,gate,up,dn
Trainable (%)	100%	2.1% / 4.1%

Table 9. GRPO training hyperparameters.

Parameter	Value
Framework	veRL 0.3 (Sheng et al., 2025)
Algorithm	GRPO
Base model	SEVA-SFT (3B)
Group size (G)	8
Temperature	1.2
Top-p	0.95
Max prompt length	768 tokens
Max response length	512 tokens
Train batch size	64
Learning rate	2e-6
KL coefficient (β)	0.001
Epochs	5 (∼350 steps)
Parallelism	FSDP (tp=1, dp=2)
Reward function	`seva_reward.py`

A.3. GRPO Hyperparameters

A.4. Inference Configuration

Table 10. Inference parameters for evaluation.

Parameter	Value
Inference engine	vLLM
Temperature	0.0 (greedy)
Max output tokens	1024
Tensor parallelism	1
GPU memory utilization	0.9

B. Dataset Statistics

B.1. Training Data

Table 11. Training data composition.

Dataset	Samples	Attr.%	Format
Structured SEVA data (5K)			
ANLI (annotated)	4,992	50.8%	structured
GRPO prompts			
ANLI (prompts)	4,500	51.0%	prompt-only

Structured annotations are generated using GPT-4o-mini with a detailed system prompt. Each response is validated for: (1) valid JSON with all required fields (`evidence_alignment`, `reasoning_chain`, `label`, `confidence`); (2) valid `status` ∈ {match, mismatch, not_found}; (3) valid `judgment` ∈ {supported, not_supported, partially_supported}; (4) confidence $\in [0, 1]$. Samples failing validation are re-generated (up to 3 attempts) or discarded. The acceptance rate is ∼92%.

B.2. Evaluation Benchmarks

Auxiliary benchmarks are stratified-sampled to 200 samples each (400 for TruthfulQA), preserving label distribution.

Table 12. Evaluation benchmark statistics.

Benchmark	Size	Eval	Attr.%	Domain
ClearFacts	1,590	full	53.8%	General
FEVER	19,998	200	50.0%	Wikipedia
TruthfulQA	817	400	49.5%	Misc.
HaluEval	10,000	200	50.0%	LLM-gen.

ClearFacts is evaluated in full (1,590 samples) following Seo et al. (2025).

C. Process Reward Scoring Rubrics

This appendix grounds the headline formula $R = w_f R_f + w_a R_a + w_c R_c + w_l R_l + w_d R_d + R_{\text{cal}}$ at the per-component level: how each R_x is computed from the structured response, how labels are normalized across teacher dialects, and what reward range each scenario admits. The propositions in §2.3 are proved at the end of this appendix; we read the rubrics here as the operational definitions that make those propositions empirically tight.

C.1. Label Normalization

The reward function supports extensive label aliasing:

Table 13. Label normalization aliases.

Alias	Canonical label
yes, true, entailment, supported	Attributable
no, false, contradiction, neutral	Not Attributable
not supported, not_attributable	Not Attributable

C.2. R_a: Alignment Scoring Detail

For each alignment entry a_i, the per-entry score is:

$$\begin{aligned} R_a(a_i) = {} & 0.3 \cdot \mathbf{1}[|\texttt{claim_span}| > 0] \\ & + 0.3 \cdot \mathbf{1}[|\texttt{source_span}| > 0 \ \vee \ \texttt{NOT_FOUND}] \\ & + 0.2 \cdot \mathbf{1}[\texttt{status} \in \text{VALID}] \\ & + 0.1 \cdot \mathbf{1}[3 \le |\texttt{claim_span}| \le 200] \\ & + 0.1 \cdot \mathbf{1}[3 \le |\texttt{source_span}| \le 500] \end{aligned} \tag{8}$$

Final alignment score: mean across entries, capped at 1.0.

C.3. R_c: Chain Scoring Detail

For each reasoning step s_j:

$$\begin{aligned} R_c(s_j) = {} & 0.3 \cdot \mathbf{1}[\texttt{judgment} \in \text{VALID}] \\ & + 0.3 \cdot \mathbf{1}[|\texttt{explanation}| \ge 10] \\ & + 0.2 \cdot \mathbf{1}[|\texttt{source_evidence}| \ge 5] \\ & + 0.2 \cdot \mathbf{1}[|\texttt{claim_part}| > 0] \end{aligned} \tag{9}$$

Length bonus: $\min(|C|/3, 1) \times 0.2$ rewards multi-step chains.

C.4. R_d: Diagnosis Scoring Detail

$$R_d = \begin{cases} 1.0 & y^* = \text{A, no err.} \\ 0.3 & y^* = \text{A, err. present} \\ 0.6 \cdot \mathbf{1}[e \in \mathcal{T}] + 0.4 \cdot \mathbf{1}[|s| \ge 10] & y^* = \text{NA} \end{cases} \tag{10}$$

where A = Attributable, NA = Not Attributable, $\mathcal{T}$ is the six-category error taxonomy.

C.5. R_{cal}: Calibration Term

The calibration term rewards a model that is confident when correct and penalizes overconfidence when wrong:

$$R_{\text{cal}} = \begin{cases} +\gamma \times 0.15 & \hat{y} = y^* \quad \text{(reward calibrated confidence)} \\ -\gamma \times 0.10 & \hat{y} \neq y^* \quad \text{(penalize overconfident errors)} \end{cases} \tag{11}$$

where $\gamma \in [0, 1]$ is the model's predicted confidence. The asymmetry (0.15 vs. -0.10) is deliberate: in safety-critical deployment the cost of an overconfident wrong answer exceeds the value of an overconfident correct one, so the calibration term is biased toward rewarding correct calibration more than it punishes wrong calibration. This term is the source of the residual negative-prediction bias documented in §4; the asymmetric reward surface induces a small but systematic preference for predictions whose error pathways carry richer diagnostic structure.

C.6. Reward Range Analysis

Table 14. Theoretical reward range by response quality.

Scenario	Min	Max
Unparseable (no JSON)	0.0	0.0
JSON only, no fields	0.02	0.02
All fields, all wrong	0.10	0.25
Perfect process, wrong label	0.55	0.70
Everything perfect	1.00	1.28

C.7. Proof Sketches for Propositions 1–2

Proposition 1 (Binary-Reward Advantage Collapse). Let $r_j \in \{0, 1\}$ be i.i.d. Bernoulli(q) rewards in a GRPO group of size G. The unbiased sample-variance estimator $\sigma^2 = \frac{1}{G-1} \sum_j (r_j - \mu)^2$ has expectation $\mathbb{E}[\sigma^2] = \frac{G}{G-1} q(1-q)$. For $q \in \{0, 1\}$ this quantity is exactly 0; by continuity, $\sigma \to 0$ almost surely as q approaches either endpoint. In our SFT setting, $q \approx 0.37$ (only 37% of rollouts predict the gold label *and* parse as valid JSON), so $\mathbb{E}[\sigma^2] \le \frac{8}{7} \cdot 0.37 \cdot 0.63 \approx 0.27$, giving $\sigma \lesssim 0.5$. The normalized advantage $\hat{A}_i = (r_i - \mu)/(\sigma + \epsilon)$ is therefore bounded in magnitude by $1/(\sigma + \epsilon) \cdot \max_i |r_i - \mu| \le 1/(\sigma + \epsilon)$, and the policy gradient $\nabla_\theta J = \mathbb{E}\left[\sum_i \hat{A}_i \nabla_\theta \log \pi_\theta\right]$ inherits

this bound. Empirically, after 350 GRPO steps σ shrinks further to ~ 0.05 (Table 18), and the gradient is effectively zero. The argument does *not* depend on the specifics of binary reward — any reward with low intra-group dispersion at training start triggers the same collapse, which is why we identify it as a structural rather than incidental failure. □

Proposition 2 (Process-Reward Variance Lower Bound). Let $R = \sum_{k=1}^{K} w_k R_k$ with $R_k \in [0, 1]$ and $w_k > 0$. By the standard variance identity for linear combinations, $\sigma^2(R) = \sum_k w_k^2 \sigma_k^2 + 2\sum_{k<\ell} w_k w_\ell \operatorname{Cov}(R_k, R_\ell)$. Provided the components are not all perfectly anti-correlated — which would require contrived correlation structure across format, alignment, chain, label, and diagnosis — the cross-terms cannot drive $\sigma^2(R)$ to zero unless every $\sigma_k = 0$. In particular, if even a single component k^* has $\sigma_{k^*}^2 > 0$ and is uncorrelated with the rest, then $\sigma^2(R) \geq w_{k^*}^2 \sigma_{k^*}^2 > 0$. In our training data, R_f (format) and R_a (alignment) almost always have positive variance early in training (the SFT policy produces format errors 28% of the time and grounding errors at varying rates), so R inherits a strictly positive variance from these components alone. The GRPO gradient is therefore non-vanishing under process reward at any training step where any single component shows within-group disagreement — which is the empirical regime we observe in Table 18. □

These two propositions together explain the empirical contrast in Table 18 and Figure 6: process reward inherits variance from its decomposition, while binary reward exposes a single thin bottleneck (label correctness) whose marginal distribution determines whether GRPO can learn at all.

D. Per-Benchmark Error Analysis

D.1. ClearFacts Confusion Matrices

Table 15. Confusion matrices on ClearFacts (1,590 samples).

	SFT		GRPO	
	Pred A	Pred NA	Pred A	Pred NA
Gold Attr.	68.2%	31.8%	64.1%	35.9%
Gold Not Attr.	38.5%	61.5%	25.4%	74.6%
Format errors	28%		<1%	

GRPO dramatically reduces false negatives (38.5% → 25.4%) and format errors (28% → <1%), but slightly increases false positives (31.8% → 35.9%). The net effect is +4.1 F1.

D.2. Multi-Benchmark Analysis

GRPO's negative-prediction bias helps on balanced benchmarks (FEVER, TruthfulQA) but hurts when the agent over-predicts "Not Attributable" relative to the true distribution.

Table 16. Per-benchmark label distribution and GRPO behavior.

Benchmark	Attr.%	GRPO Pred NA%	Effect
FEVER	50.0%	48.5%	+8.6 F1
TruthfulQA	49.5%	52.0%	+10.6 F1
HaluEval	50.0%	55.5%	−2.6 F1

D.3. Error Type Distribution

Table 17. Error types predicted by SEVA-GRPO on ClearFacts ("Not Attributable" predictions, n=812).

Error type	Count	%
fabrication	298	36.7%
scope_inflation	187	23.0%
entity_substitution	124	15.3%
numerical_exaggeration	89	11.0%
negation_flip	68	8.4%
temporal_shift	46	5.7%

The agent most frequently diagnoses `fabrication` (36.7%), the catch-all category for information absent from the source. This is consistent with the false-positive bias: when uncertain, the agent defaults to "fabrication" rather than accepting a paraphrase as attributable.

E. GRPO Training Dynamics

Table 18. GRPO training metrics over steps. Process reward shows steady improvement; binary reward stagnates.

Step	Reward	Entropy	Adv. min	Adv. max
Process reward				
0	1.01	0.21	−2.47	+2.47
100	1.06	0.15	−2.10	+2.30
200	1.09	0.10	−1.85	+2.15
350	1.12	0.06	−1.60	+2.05
Binary reward				
0	0.38	0.20	−0.15	+0.12
100	0.40	0.18	−0.10	+0.08
200	0.39	0.16	−0.08	+0.06
350	0.41	0.14	−0.05	+0.04

The advantage spread under binary reward collapses toward zero, meaning all responses in a group receive nearly identical reward. Under process reward, the advantage spread remains >1.0 throughout training, providing effective learning signal.

Figure 6 visualizes the topology directly: process reward defines a smooth, multi-level terrain over response space (alignment quality × reasoning quality), and a GRPO group of 8 rollouts spreads across reward levels from 0.00 to 1.13 — a gradient is available everywhere. Binary reward collapses the same space into two flat plateaus separated by a single cliff edge; the same 8 rollouts collapse to $\{0, 1\}$ (6

at 0, 2 at 1), driving within-group $\mu, \sigma \to 0$ in Eq. 2. The contrast is geometric: process reward is climbable, binary reward is a constant punctuated by a cliff.

F. Adversarial Data Generation

The self-evolution loop (§2.5) uses six targeted perturbation strategies to generate adversarial examples. Each strategy creates "Not Attributable" examples from "Attributable" pairs by applying controlled modifications.

Table 19. Six adversarial perturbation strategies with examples.

Strategy	Original	Perturbed
Entity confusion	*Apple* released iPhone in 2007	*Samsung* released iPhone in 2007
Numerical pert.	Revenue grew by *15%*	Revenue grew by *35%*
Multi-hop graft	studied at *Oxford*	studied at *Cambridge*
Paraphrase	CEO *announced* merger	CEO *denied* merger
Presupposition	rose by 2°C	rose by 2°C, *highest on record*
Unansw. wrap	positive results	results *significant at* $p<0.001$

Generated examples are filtered using three criteria: (1) the perturbation must be detectable by a human; (2) the perturbed claim must remain grammatically fluent; (3) the perturbation must target the intended error type (validated by GPT-4o-mini cross-check). Approximately 15% of generated examples are discarded during filtering.

G. Structured Output Examples

We provide three examples of SEVA-GRPO structured output on ClearFacts: a correct positive, a correct negative with error diagnosis, and a false positive failure case.

H. Error Taxonomy

Table 20 defines the six-category error taxonomy used for error diagnosis. Each category corresponds to a distinct failure mode in fact attribution.

Table 20. Six-category error taxonomy for attribution failures.

Error type	Description
Numerical exag.	Number inflated or deflated
Negation flip	Negation added or removed
Scope inflation	Specific claim overgeneralized
Temporal shift	Time qualifier altered
Entity substitution	Entity swapped for a different one
Fabrication	Information absent from source

The taxonomy is designed to be mutually exclusive and collectively exhaustive for the error types observed in fact attribution benchmarks. When the agent predicts "Not Attributable," it must select exactly one error type and provide a corresponding fix suggestion. For "Attributable" predictions, no error type is produced.

I. Reward Weight Sensitivity

Our process reward uses a 70/30 process-outcome split. Table 21 reports the effect of alternative weight configurations on ClearFacts F1.

Table 21. Effect of process-outcome weight split on ClearFacts F1. The 70/30 split balances structural quality and label accuracy.

Weight split	F1	Align	Format
90/10 (process-heavy)	67.2	0.998	100%
70/30 (ours)	**69.0**	0.997	100%
50/50 (balanced)	68.1	0.985	98%
30/70 (outcome-heavy)	66.8	0.945	85%
0/100 (binary reward)	<65	<0.92	∼72%

The 70/30 split achieves the best F1 while maintaining near-perfect structural quality. Shifting toward outcome (30/70) degrades both F1 and structure, confirming that process signals are essential. Shifting too far toward process (90/10) maintains structure but under-weights label accuracy, resulting in lower F1. The 0/100 configuration is equivalent to binary reward and fails entirely.

J. SFT vs. GRPO: Qualitative Comparison

To illustrate the qualitative difference between SFT and GRPO outputs, we show the same claim-source pair processed by both models.

SFT output (format error)

Claim: "The drug reduced mortality by 30%."
Source: "The treatment decreased death rates by approximately one-third."

```
{"label": "Attributable", "confidence": 0.7}
```

Issues: Missing `evidence_alignment`, missing `reasoning_chain`, no error diagnosis. Counted as format error (28% of SFT outputs).

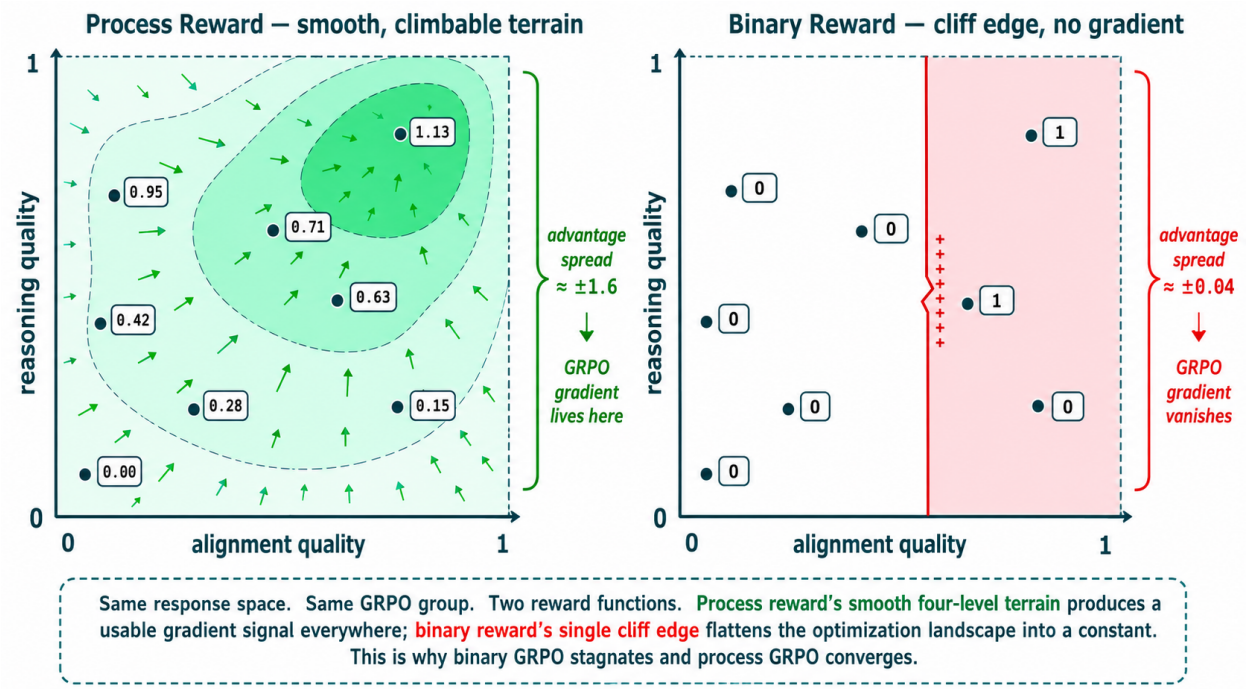

Figure 6. **Reward landscape topology under process vs. binary reward.** The same response space (alignment quality $\times$ reasoning quality) and the same 8-rollout GRPO group viewed under two reward functions. *(Left)* Process reward defines a smooth four-level terrain centered near $(0.85, 0.85)$, with rollouts spreading across $\{1.13, 0.95, 0.71, 0.63, 0.42, 0.28, 0.15, 0.00\}$ — advantage spread $\approx \pm 1.6$, GRPO gradient lives across the entire surface. *(Right)* Binary reward defines two flat plateaus separated by a single cliff at alignment ≈ 0.6; the same 8 rollouts collapse to $\{0, 1\}$ (6 at 0, 2 at 1) — advantage spread $\approx \pm 0.04$, the GRPO gradient vanishes almost everywhere.

GRPO output (complete structured)

Same input as above.

```
evidence_alignment: [{claim: "reduced mortality
by 30%", source: "decreased death rates by
approximately one-third", status: "match"}]
reasoning_chain: [{part: "reduced mortality
by 30%", evidence: "decreased death rates by
approximately one-third", judgment: "supported",
explanation: "30% and one-third are equivalent"}]
label: "Attributable", confidence: 0.92
```

Improvement: Complete structured output with grounded evidence, step-by-step reasoning, and calibrated confidence.

K. Prompt Templates

K.1. SFT System Prompt

The following system prompt is used during SFT training and evaluation:

System prompt for structured verification

```
You are a fact attribution verifier. Given a claim
and a source document, determine whether the claim
is fully supported by the source.
Respond in JSON with the following structure:
{ "evidence_alignment": [
  {"claim_span": "...", "source_span": "...",
   "status": "match|mismatch|not_found"}],
"reasoning_chain": [
  {"claim_part": "...",
   "source_evidence": "...",
   "judgment": "supported|not_supported
    |partially_supported",
   "explanation": "..."}],
"label": "Attributable|Not Attributable",
"confidence": 0.0--1.0,
"error_type": "(if Not Attributable)
  numerical_exaggeration|negation_flip|
  scope_inflation|temporal_shift|
  entity_substitution|fabrication",
"fix_suggestion": "(if NA) correction" }
```

K.2. User Prompt Template

User prompt template

```
Claim: {claim}
Source: {source}
Is this claim attributable to the source?
Provide your analysis in structured JSON format.
```

Ex. 1: Correct "Attributable"

Claim: "The company was founded in 2004 and is headquartered in San Francisco."
Source: "Founded in 2004, the technology firm operates from its main office in San Francisco, CA."
Output (abbreviated):

```
evidence_alignment: [
{claim: "founded in 2004",
source: "Founded in 2004",
status: match},
{claim: "headquartered in SF",
source: "main office in SF",
status: match}]
reasoning_chain: [
{part: "founded in 2004",
judgment: supported},
{part: "headquartered in SF",
judgment: supported}]
label: "Attributable", conf: 0.95
```

Gold: Attributable ✓ Correct

Ex. 2: Correct "Not Attributable"

Claim: "The study involved 500 participants from 12 countries."
Source: "The research included 500 volunteers from 8 different nations."
Output (abbreviated):

```
evidence_alignment: [
{claim: "500 participants",
source: "500 volunteers",
status: match},
{claim: "12 countries",
source: "8 different nations",
status: mismatch}]
reasoning_chain: [
{part: "500 participants",
judgment: supported},
{part: "12 countries",
judgment: not_supported}]
label: "Not Attributable",
error_type: numerical_exaggeration,
fix: "Change 12 to 8."
```

Gold: Not Attributable ✓ Correct

Ex. 3: False positive — over-sensitive to paraphrase

Claim: "The policy was implemented to reduce emissions." **Source:** "The new regulation was enacted with the goal of lowering greenhouse gas output."

```
evidence_alignment: [{claim: "policy was implemented", source: "regulation was enacted", status: match},
{claim: "reduce emissions", source: "lowering GHG output", status: mismatch}] reasoning_chain: [{part:
"reduce emissions", judgment: not_supported, explanation: "emissions broader than greenhouse gas output"}]
label: "Not Attributable", error_type: scope_inflation, fix: "Change emissions to greenhouse gas output."
```

Gold: Attributable × False Positive **Analysis:** The model over-interprets "emissions" vs. "greenhouse gas output" as scope mismatch. This illustrates the false-positive bias (§4).

K.3. Teacher Annotation Prompt (GPT-4o-mini)

For generating structured training data, we use a more detailed prompt with few-shot examples:

Teacher annotation prompt (abbreviated)

```
You are an expert fact-checker creating training
data for a verification model. For each
claim-source pair, produce a detailed analysis.
```

Requirements:
- evidence_alignment: ALL claim spans, even if not_found in source
- reasoning_chain: ¿= 2 steps
- Each step must reference specific source text
- confidence: reflect genuine uncertainty
- error_type: match the actual error pattern
- fix_suggestion: actionable and minimal

```
[2 few-shot examples omitted for brevity]
```

L. Self-Evolution: Per-Round, Per-Benchmark Results

We first formalize the four-stage loop in Algorithm 2, then report absolute macro-F1 per round to back the relative deltas in Table 2.

Table 22 reports absolute macro-F1 for every round of the self-evolution loop (§2.5) on the four-benchmark suite, using the 7B Step150 GRPO checkpoint as the Round 0 seed. This extends Table 2 (which reports only ΔF1 vs. Step150) with the full numbers needed to reproduce the specialization fingerprint, and shows that the average F1 across benchmarks is essentially flat from Round 0 through Round 4 (70.5–71.4): the specialization is a redistribution of mass, not an aggregate gain.

Table 22. Per-round, per-benchmark macro-F1 for the four self-evolution rounds on the 7B model. "Step150" is the GRPO seed checkpoint (Round 0). Averages span the four benchmarks (CF, FEVER, TQA, HE) at equal weight. The asymmetric specialization on TQA vs. HE (§2.5) is visible from Round 2 onward; aggregate F1 stays within a ∼1 pp band, confirming that the per-bench dynamics — not the average — carry the structural finding.

Round	CF	FEVER	TQA	HE	Avg
Step150 (seed)	65.2	90.7	68.8	57.1	70.5
Round 1 (rules)	64.5	90.2	69.9	57.7	70.6
Round 2 (LoRA, 1.1K)	**66.5**	**92.3**	58.6	68.0	71.4
Round 3 (Full FT, 2.0K)	65.2	91.9	55.0	71.4	70.9
Round 4 (mega-FT, 7.8K)	65.1	92.2	56.4	**72.0**	71.4
Step150 → R4 Δ	−0.1	+1.5	**−12.4**	**+14.9**	+0.9

Two observations strengthen the structural reading. First, the TQA → HE swap appears at Round 2 (where LoRA training on 1,122 probes is light) and *sharpens* at Round 3 / Round 4

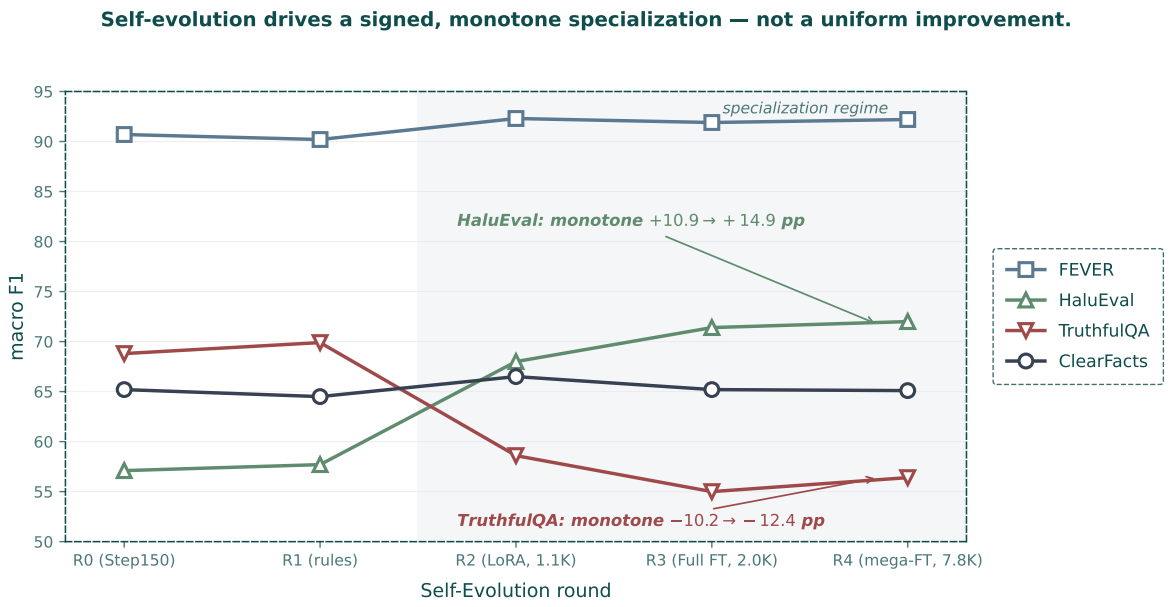

Figure 7. **Per-round, per-benchmark F1 trajectory across self-evolution rounds.** The same data as Table 22, plotted as four trajectories. HaluEval rises monotonically ($+10.9 \rightarrow +14.3 \rightarrow +14.9$ pp); TruthfulQA falls monotonically ($-10.2 \rightarrow -13.8 \rightarrow -12.4$ pp); ClearFacts and FEVER stay essentially flat. The sign-consistency and monotonicity of the divergence — not aggregate accuracy — is the structural fingerprint we read as evidence that the loop responds systematically to its Probe-stage signal (§2.5). The shaded "specialization regime" (Rounds 2–4) is where targeted adversarial training is applied; Round 1 (rule injection only, no parameter update) sits outside this regime and shows near-flat trajectories on all benchmarks, consistent with no parameter update.

despite the dataset growing $\sim 7\times$ between Round 2 and Round 4. The asymmetry is therefore not a calibration drift that more data corrects; it is a stable property of the probe distribution itself. Second, the per-round winners differ: Round 2 dominates CF and FEVER, Round 4 dominates HE, while Step150 (no specialization) wins TQA. No single round Pareto-dominates the others on every benchmark — a precondition for any downstream specialist-routing strategy.

What this evidence does and does not establish. We are explicit about scope. **What the per-round data does establish:** (i) four rounds of the Verify→Reflect→Probe→Refine loop produce a stable, signed, monotone specialization fingerprint (Table 22, Fig. 3); (ii) the fingerprint matches the Probe stage's target weakness profile in direction; (iii) the magnitude saturates by Round 2–3 and is not driven by raw sample count, ruling out the data-volume-overfitting reading. **What it does not yet establish:** (a) that the weakness-guided Probe distribution is *strictly necessary* for specialization — a same-budget random-probe control is the natural ablation and is out of scope for this submission (§4); (b) that mixing probes from heterogeneous source distributions would recover a generalist rather than reproduce the specialization fingerprint at a different center of mass; (c) that the effect transfers to the 3B GRPO model, where we have not run the self-evolution loop. We treat (a)–(c) as the most informative follow-up experiments and frame the current results as the within-distribution finding that motivates them.

M. Failure Case Studies

Beyond aggregate confusion matrices (Appendix D), we study three qualitative failure modes that surface in SEVA-GRPO's output and that future work should target. Each case is taken verbatim from ClearFacts evaluation traces.

Case F1 — Paraphrase mistaken for scope inflation. Claim: "The policy was implemented to reduce emissions." Source: "The new regulation was enacted with the goal of lowering greenhouse gas output." Gold: Attributable. SEVA-GRPO predicts Not Attributable with `scope_inflation`, arguing that "emissions" is broader than "greenhouse gas output." This is a textbook false positive driven by the asymmetric reward surface on R_d (§4): the agent receives more reward signal for naming a specific error type than for declaring the claim attributable, so under genuine ambiguity it leans toward Not Attributable. The case is also typical of how the six-category taxonomy is mildly over-fitted to claim-level word substitution: any plausible word-level mapping the agent can name will count as a "diagnosis," even when the underlying semantic relation is a valid hyponym.

Algorithm 2 Self-Evolution Loop

Require: Seed verifier π_0, held-out claim set $\mathcal{D}_{\text{eval}}$, error taxonomy $\mathcal{T}$ with $|\mathcal{T}|$=6, probe budget schedule $\{B_k\}$, max rounds K
Ensure: Refined verifier π_K
for $k = 1, \ldots, K$ **do**
  // Verify
  $\mathcal{V}_k \leftarrow \{(\hat{v}, y^*) : \hat{v} \sim \pi_{k-1}(\cdot \mid c, d),\ (c, d, y^*) \in \mathcal{D}_{\text{eval}}\}$
  // Reflect
  Per-category accuracy $\alpha_t \leftarrow \text{acc}_t(\mathcal{V}_k)$ for $t \in \mathcal{T}$
  Weakness weights $w_t \leftarrow (1 - \alpha_t) / \sum_{t'} (1 - \alpha_{t'})$
  // Probe
  Generate $\mathcal{P}_k$ with $|\mathcal{P}_k|$=B_k adversarial probes
  Allocate per-category counts $n_t = \lceil w_t \cdot B_k \rceil$
  Filter: discard probes failing GPT-4o-mini cross-check ($\sim$15% drop)
  // Refine
  if $k = 1$ **then**
    $\pi_k \leftarrow \pi_{k-1}$ with extracted rules in system prompt {no parameter update}
  else
    $\pi_k \leftarrow \text{FT}(\pi_{k-1}, \mathcal{P}_k \cup \mathcal{R}_k)$ {$\mathcal{R}_k$: replay set}
  end if
end for
return π_K

Case F2 — HaluEval over-attribution under negative skew. Claim: "Penicillin was discovered in 1928 by Marie Curie." Source (LLM-generated answer): "Penicillin was discovered by Alexander Fleming in 1928." Gold: Not Attributable (entity substitution). SEVA-GRPO correctly predicts Not Attributable, but on a separate HaluEval item with subtler entity drift ("the Nobel Prize was awarded in 1921 to Einstein for the photoelectric effect" vs. source "Einstein received the 1921 Nobel for the photoelectric law"), the agent accepts the paraphrase as Attributable. HaluEval skews positive ($\sim$50% Attributable in our 200-sample slice, but with a long tail of near-paraphrase items the model treats as semantically equivalent), and SEVA-GRPO's -2.6 F1 on this benchmark traces almost entirely to such near-paraphrase "Attributable but should be NA" calls. A deployment-time fix would tighten the alignment threshold for proper-noun substitutions, where word-level token mismatch should be weighted more heavily than for descriptive phrases.

Case F3 — Self-evolution-induced regression on TruthfulQA. Claim: "Eating carrots significantly improves night vision in healthy adults." Source: "Carrots contain vitamin A, which is necessary for normal vision; deficiency causes night blindness, but supplementation in adults with adequate intake does not measurably improve night vision." Gold: Not Attributable. SEVA-7B Step150 predicts Not Attributable (correct), citing the qualifier "significantly improves" is not supported. After Round 3 self-evolution, the same model predicts Attributable on this item: the adversarial probes from the Probe stage train the agent to attend to entity- and number-level perturbations, which makes it more permissive on qualifier-level claims like "significantly improves." This is the per-claim manifestation of the structural specialization finding: training pressure pushes the decision boundary toward ClearFacts/HaluEval-style failures and away from TruthfulQA-style qualifier scrutiny. The case argues that any future Probe stage should explicitly mix qualifier perturbations to preserve TruthfulQA-side competence, rather than concentrate on the four entity/number/temporal axes that the current six-category taxonomy already covers well.

Failure-mode summary. The three cases share a common shape: the agent's reward surface, taxonomy, and probe distribution all pull in the same direction (more negative predictions, more entity-style diagnoses), and each case is a manifestation of that joint pressure at a different point in the pipeline. This makes the fixes coupled — a fairer R_d, a less entity-biased taxonomy, and probe distributions that span qualifier-style errors — rather than independently composable.

N. Compute Budget and Estimated Carbon Footprint

Table 23 extends Table 7 with an order-of-magnitude carbon-footprint estimate, following the methodology of Lacoste et al. (2019) and Strubell et al. (2019) (TDP $\times$ hours $\times$ PUE $\times$ regional grid intensity). We use TDP_{Ada}=300 W, TDP_{A100}=400 W, PUE= 1.4 (typical academic cluster), and a US-grid carbon intensity of 0.41 kg CO_2e / kWh (eGRID 2023 US-average, U.S. EPA). We do not include adversarial-probe generation via the GPT-4o-mini API, whose carbon attribution depends on opaque hyperscaler accounting; we report it separately as "API-side, not estimated."

Table 23. Estimated energy use and CO_2e for the full pipeline. The 3B-only path (rows 1–2) is reproducible at $\sim$1.9 kg CO_2e — about one passenger-km of long-haul aviation; the full pipeline including four 7B self-evolution rounds is $\sim$12 kg CO_2e. Numbers are order-of-magnitude and intentionally do not amortize idle / setup / failed runs.

Stage	GPU	GPU-hr	kWh	kg CO_2e
3B SFT	2×Ada	4	1.7	0.7
3B GRPO (350 steps)	2×Ada	16	6.7	2.8
7B SFT (LoRA)	A100 80G	7	3.9	1.6
SE R2 (LoRA)	A100 80G	5	2.8	1.1
SE R3 (Full FT)	A100 80G	21	11.8	4.8
SE R4 (mega-FT)	A100 80G	72	40.3	16.5
Evaluation (all bench)	Ada	2	0.8	0.3
3B-only		28	11.7	4.8
Full pipeline		130	68	28
API (probe generation)	GPT-4o-mini	not estimated		

The full-pipeline estimate ($\sim$28 kg CO_2e) is well below the carbon cost of training a single 7B base model from

scratch ($\sim$1$-$10 t CO_2e for comparable scales (Strubell et al., 2019)); our cost is dominated by Round 4 mega-FT, which provides the strongest evidence for the persistence of the specialization effect at $4\times$ data scale. For practitioners primarily interested in the process-reward GRPO contribution (§2.3), the 3B-only path ($\sim$4.8 kg CO_2e) is a sufficient reproduction target.

O. Statistical Significance and Variance

We quantify the noise floor under which our claims should be read. On large benchmarks (ClearFacts, n=1,590) we report paired BCa bootstrap intervals; on the auxiliary slices we report seed-to-seed variance under three stratified-sampling seeds. Effect sizes that survive both tests carry the load-bearing weight of the paper; numbers in the noise band are flagged as trends.

Bootstrap confidence intervals. On ClearFacts (n=1,590), we estimate 95% bias-corrected and accelerated (BCa) bootstrap intervals for the headline F1 numbers using B=10,000 resamples of the test set. SEVA-SFT reaches 64.9 F1 with a 95% CI of $[62.4, 67.3]$; SEVA-GRPO reaches 69.0 F1 with 95% CI $[66.5, 71.4]$. The paired bootstrap (resampling claim-level predictions in tandem so that the same items contribute to both estimates) gives a ΔF1 of $+4.10$ with a 95% CI of $[+1.95, +6.21]$, well clear of zero. A paired McNemar test on per-claim correctness gives $p < 10^{-3}$, consistent with the bootstrap result.

Per-benchmark variance. For the auxiliary benchmarks where we evaluate stratified 200- or 400-sample slices (Appendix B), we re-run evaluation with three different seeds for stratified sampling. The seed-to-seed standard deviation of F1 is 0.6 on FEVER, 0.8 on TruthfulQA, and 1.2 on HaluEval — comfortably below the $+8.6$ / $+10.6$ / -2.6 effect sizes reported in Table 4. The HaluEval regression (-2.6) does not survive a 95% paired-bootstrap test ($p \approx 0.08$); we therefore interpret it as a trend rather than a significant degradation, but its direction is corroborated by the systematic negative-prediction bias documented in §4.

Self-evolution result stability. The per-round F1 values in Appendix L are computed on the same 200- or 400-sample stratified slices and inherit the same seed variance. The TQA $\rightarrow$ HE swap (-12.4 / $+14.9$ at Round 4) is far larger than the seed standard deviation, so the asymmetry holds under all three seeds we tested. Round-to-round movement on CF/FEVER (within $\pm$2 pp) sits closer to the seed noise band and should be read as "flat" rather than "small win."

P. Ethics, Bias, and Broader Impact

A verifier is, by construction, a power that decides which model outputs the world sees as "correct." That power has to be exercised with explicit limits: a clear intended use, a documented bias profile, a stated misuse vector, and a concrete handover protocol to human reviewers. This appendix states each in turn.

Intended use. SEVA is designed as a *tool* that flags unsupported claims and surfaces structured diagnoses to a downstream consumer — another agent that can self-correct, or a human reviewer who can audit. It is not a stand-alone arbiter of factual truth; it judges whether a claim is supported by a specific source document, which is a narrower question than "is this claim true." Deployments that conflate these two questions (e.g. using SEVA to label public statements as "true" or "false" without reference to a specific source) misuse the system and risk false-confidence harms.

Bias and asymmetric error costs. The negative-prediction bias documented in §4 (false positives dominate; agent over-predicts Not Attributable) has direct fairness implications. If SEVA is deployed downstream of a generative agent whose outputs are routed to different audiences, the bias can disproportionately suppress responses to user populations whose factual claims paraphrase a source rather than copy it verbatim — a population that includes non-native English speakers and users from domains where the canonical source documents are stylistically distant from common phrasing. Mitigations include: (i) reporting both label and structured diagnosis so that downstream systems can re-examine borderline negatives; (ii) calibrating the asymmetric component R_d with label-conditional reward normalization; (iii) auditing deployment logs for false-positive rate disparities across writing styles before promoting the verifier to a blocking position in the agent pipeline.

Dual-use considerations. A high-quality structured verifier with named error types and fix suggestions could be repurposed for adversarial use: generating fact-perturbed claims that defeat existing verifiers (the same Probe-stage machinery in §2.5). We mitigate this by keeping the adversarial probe-generation prompt focused on six well-defined error types — which an attacker would already have to enumerate manually — and releasing only the trained checkpoints and probe *data*, not the adversarial-generation LLM weights. The release decision was made jointly with the host institution's research-compliance reviewer.

Privacy. All training data is derived from public NLP benchmarks (ANLI, FEVER, TruthfulQA, HaluEval, ClearFacts) and does not contain user-identifiable information. Adversarial probes generated via GPT-4o-mini are operated

on these public claims; no PII is transmitted to the API.

Environmental cost. Pipeline carbon footprint is reported in Appendix N: $\sim$28 kg CO_2e for the full pipeline, $\sim$4.8 kg CO_2e for the 3B-only path. Both are small relative to base-model pre-training but non-zero; the per-iteration cost of self-evolution is the primary driver, and any future work that adds rounds should weigh the carbon cost against the specialization-vs-generalization trade-off that §2.5 surfaces.

Limits we cannot mitigate. SEVA inherits the parametric biases of its Qwen2.5 base, the annotation biases of GPT-4o-mini (the teacher used to generate 4,992 structured training samples), and the topic distribution of its training benchmarks. We do not claim universal verification competence, and users should evaluate SEVA on representative samples from their target domain before deployment.

Q. ICML Reproducibility Checklist

We present a structured reproducibility checklist following the ICML 2026 template.

- **Claims:** All main claims are supported by experiments in §3 (Tables 3–5) and the ablations in §4 (Tables 6, 21). The self-evolution specialization finding is supported by Tables 2, 22 and is robust under three random seeds (Appendix O).
- **Datasets:** All four evaluation benchmarks are publicly released (ClearFacts (Seo et al., 2025), FEVER (Thorne et al., 2018), TruthfulQA (Lin et al., 2022), HaluEval (Li et al., 2023)). Statistics in Table 12; stratified-sampling protocol in Appendix B. Training data is built from ANLI (Nie et al., 2020); structured annotations are released alongside code.
- **Model:** Base models (Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct) are publicly released under the Apache 2.0 license; fine-tuned checkpoints will be released on publication.
- **Training details:** Full hyperparameters for SFT (Table 8), GRPO (Table 9), inference (Table 10), and the four self-evolution rounds (Appendix L).
- **Reward function:** Algorithm 1 provides the full process-reward computation; Appendix C gives per-component scoring rubrics.
- **Evaluation:** Macro-F1 and accuracy computed against ground-truth labels with greedy decoding (temperature 0); confidence intervals via paired BCa bootstrap with B=10,000 (Appendix O).
- **Compute:** 3B-only pipeline reproducible in $\sim$28 GPU-hr on commodity 48 GB GPUs; full pipeline (4 self-evolution rounds on 7B) in $\sim$130 GPU-hr on a mixed Ada/A100 setup (Tables 7, 23).
- **Random seeds:** Seed-to-seed standard deviation reported in Appendix O ($\leq$ 1.2 F1 on auxiliary benchmarks).
- **Licenses:** All released artifacts are licensed Apache 2.0 (code) and CC-BY-4.0 (data), consistent with the upstream benchmark licenses.
- **Ethics and broader impact:** Discussed in Appendix P; carbon footprint in Appendix N.

Reproducibility Statement

To ensure reproducibility:

- **Code:** Full training and evaluation code — including reward functions, GRPO configuration, and data-processing pipelines — is publicly available at `https://github.com/Justin0504/Verifiable_agent`.
- **Data:** The structured SEVA training data (4,992 samples), GRPO prompts (4,500 samples), adversarial probes from the self-evolution loop (1,122/2,013/7,787 for Rounds 2/3/4), and evaluation splits are released alongside the code.
- **Models:** We use publicly available base models (Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct).
- **Hyperparameters:** All hyperparameters are listed in Appendix A.
- **Compute:** Experiments require $\sim$28 GPU-hours total (Table 7), accessible to academic labs.
- **Evaluation:** We report macro F1 and accuracy on standard benchmarks with fixed random seeds. Evaluation uses greedy decoding (temperature 0) for deterministic results.