
Reinforcement Learning from Verifiable Feedback: Teaching Language Models to Say “I Don’t Know”

Eduardo Perez Denadai
Optimality.ai

Abstract

Reinforcement learning from verifiable feedback is usually studied on one skill and scored by pass rate; production needs three heterogeneous behaviours at once, including the one that matters most in deployment—declining what the facts do not support. We post-train a single 1.5B parameter code model across three verifiably distinct tasks with RLVF: proof carrying with honest abstention, intent fidelity with findings, and action routing with named refusals. Every label is computed by an exact oracle—a proof kernel, a fidelity audit, a closed policy—nothing is hand-labelled, and eight gates were registered before training. Across nine configurations—data dose, checkpoints, sequential refreshes, learning-rate schedule, weight soup—the official candidate passes **seven**: intent 90.9, router 81.0, ESCALATE recall 84.7, decidable-suite yield 97.8, 0 faults, and a clean plain-prompt serving battery, with the single residual on the held-out undecidable gap family (15.2% overclaim against the registered $\leq 5\%$). The nine probes do more than select a candidate—they map one exchange surface at this capacity: closing the gap family to zero overclaim costs intent along a measured rate ($90.9 \rightarrow 65.5$ against $15.2 \rightarrow 0$), so the residual is a capacity-point property of the surface rather than an artifact of the training recipe, and the full per-configuration record lives in an appendix rather than in prose. A premise-fidelity probe from that appendix exposed and closed a laundering channel no artifact-level oracle can see: invented preconditions $7 \rightarrow 0$; undecidable-side overclaim dropped from 26.5% to 0.0%, and recall reached 100% under a pre-registered fidelity-aware reward—a sound oracle is not sufficient, the reward must see the task-level obligation. Serving-side, the candidate fails *closed* when deactivated (10/10 declines), and 8-bit quantization reproduces the full gate suite at parity (7/8) at $1.8\times$ decode. Our findings characterize the Pareto frontier specifically for a single 1.5B architecture under single-seed training protocols.

1 Introduction

Reinforcement learning from verifiable feedback is usually studied on one skill at a time—a math domain, a coding domain—with pass rate as the score; the RLVR/GRPO line makes that scalar work harder through denser credit, better baselines, and teacher guidance [Shao et al., 2024, Guo et al., 2025, Tiwari et al., 2026]. Production systems need three heterogeneous behaviours at once, and, more importantly, they need the *negative* behaviour: declining what the facts do not support. We ask a narrower and more measurable question: given three components that each have an exact oracle, does joint post-training on all three *compose* one fail-closed policy, or do the skills interfere?

The three components and their oracles:

component	skill	oracle
prover	proof-carrying code or ABSTAIN	proof kernel (<code>certify</code>)
intent	BUILD/ASK/blocked + findings	fidelity audit (<code>audit</code>)
router	ROUTE/ESCALATE/ABSTAIN + reason	closed policy (<code>decision_dict</code>)

Every training and evaluation label is oracle-derived; nothing is hand-labelled. The oracles operate offline during training: every candidate completion is disposed by its component’s oracle, verdicts that fail the typed reward table ρ (including the laundering detectors) are excluded, and only oracle-consistent completions become training targets; at evaluation, the same oracles dispose the model’s outputs. The hypothesis: all three teach the *same* meta- behaviour—decide only what the facts support, otherwise decline and *name* what is missing—so they should compose, subject to capacity.

Contributions: (1) a harness and evaluation protocol with eight pre-registered gates spanning templated accuracy and plain-prompt serving; (2) the nine-configuration frontier: one residual, structurally mapped; (3) the reward-laundering result—a sound per-artifact oracle is not sufficient; the reward must see the task-level obligation—with a registered prediction confirmed; (4) serving and deployment findings: fail-closed deactivation, plain-prompt skill separability, and quantization parity measured by re-running the gates, not perplexity.

2 Related work

RLVR. Reinforcement learning with verifiable rewards directly optimizes task success using automatically checkable outcomes, with GRPO the standard critic-free approach for mathematical and code reasoning [Shao et al., 2024, Guo et al., 2025]. Outcome rewards, however, are sparse and coarse: GRPO assigns the same trajectory-level advantage to every token and yields no gradient when all responses in a group receive the same reward [Mishra et al., 2026, Nie et al., 2026]. The literature addresses those limitations through dynamic sampling [Yu et al., 2025], correcting optimization and length biases [Liu et al., 2025], or finer-grained credit assignment [Li et al., 2026, Jiang et al., 2025, Wang et al., 2026]; DiffGate [Tiwari et al., 2026] complements outcome rewards with dense teacher guidance where the group-relative signal is weak. All of these make the outcome *scalar* denser. RLVR asks: *how do we make a scalar success reward denser so the policy gradient flows?* Our setting asks the orthogonal question: *what happens when the oracle is exact but the reward can be gamed, the right answer is “I don’t know,” and three verifiable skills must coexist in one small model?* Density of signal is their bottleneck; honesty of signal and composition are ours. Our position between the two lines is explicit: the feedback interface is *typed* (Definition 1), the optimization used for the reported candidates is oracle-guided maximum likelihood with reward-table filtering (§3.1), and group-relative policy optimization *over typed verdict alphabets* is posed as future work (§9). The distinction is not cosmetic: a denser outcome reward is blind to premise invention—the verifier certifies the artifact—and so would a teacher-guidance channel that reshapes credit inside the same reward, which is exactly the channel our `premise_invention = −1.00` term had to open (§5).

Verifiable feedback beyond the scalar. Verifier- and process-reward training raises math and coding performance [Cobbe et al., 2021, Uesato et al., 2022, Lightman et al., 2023]; RLAI replaces human labels with model judgments [Lee et al., 2023]. Our feedback is neither prose-judged nor a pass rate: it is typed verdicts from pinned deterministic oracles, and honest abstention is *positively* scored. Learned judges are a documented failure regime—position, verbosity, and self-enhancement biases [Zheng et al., 2023] and reward hacking under specification mismatch [Skalse et al., 2023]; we make “exact-compute-or-nothing” a standing rule and measure one laundering incident when the rule was violated (§5).

Interference and composition. Multitask bias [Caruana, 1997], gradient-based forgetting mitigation [Kirkpatrick et al., 2017, Lopez-Paz and Ranzato, 2017], and model soups [Wortsman et al., 2022] frame our result. We contribute a per-class dose–response frontier at one capacity point: sequential refreshes are anti-compositional (fix shown, poison hidden), while weight averaging partially composes across trajectories—but interferes on one class (ASK), consistent with task-vector interference [Ilharco et al., 2023]. The locus of the abstention boundary also connects to selective prediction [Chow, 1957] and to the Ω -certifiability framing of claim-vs-decidable behavior [Denadai, 2026].

Efficient fine-tuning and deployment. LoRA [Hu et al., 2022] throughout; quantization judged the way it should be—by re-running the registered evaluation—rather than by perplexity, in the spirit of the quantization literature [Dettmers et al., 2022, Frantar et al., 2023, Lin et al., 2024].

3 Setup

3.1 Formal setup: typed feedback against a scalar reward

Definition 1 (Components and oracle verdicts). *Three components $k \in \mathcal{K} = \{P, I, R\}$ (prover, intent, router) share one policy π_θ over prompts $x \sim \mathcal{D}_k$. Each component has a pinned deterministic oracle $\Omega_k : \mathcal{Y} \rightarrow \mathcal{V}_k$ over a typed verdict alphabet:*

$$\mathcal{V}_P = \{\text{cert}, \text{ref}, \text{h-abstain}, \text{abstain-on-A}\}, \quad \mathcal{V}_I = \{\text{build}, \text{ask}, \text{blocked}\}, \quad \mathcal{V}_R = \{\text{route}, \text{esc}, \text{abstain}\},$$

plus the laundering detectors $\mathcal{V}' \supseteq \{\text{fabricated}, \text{premise_inv}\}$ evaluated on certified artifacts. Feedback is the typed verdict $\Omega(x, y) \in \mathcal{V}_k \cup \mathcal{V}'$, never a scalar.

For contrast, the RLVR/GRPO scalarization of the same contact surface assigns trajectory reward R_i and the group-relative advantage

$$A_i^{\text{GRPO}} = \frac{R_i - \bar{R}_G}{s_G + \varepsilon}, \quad \nabla_\theta J_{\text{GRPO}} = \frac{1}{G} \sum_{i=1}^G \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} A_i^{\text{GRPO}} \nabla_\theta \log \pi_\theta(y_{i,t} \mid x, y_{i,<t}). \quad (1)$$

Observation 1 (Vanishing group signal). *If all rewards in a group agree ($s_G = 0$), then $A_i^{\text{GRPO}} = 0$ for every i and $\nabla_\theta J_{\text{GRPO}} = \mathbf{0}$: the step contributes nothing regardless of trajectory quality. Under output-only reward every token in a trajectory also receives the identical advantage.*

Derivation. $\bar{R}_G = R$ and $s_G = 0$ give $A_i = 0$ termwise in Eq. (1); with only a terminal reward, the return at every position equals R_i , so the advantage has no t -dependence [Mishra et al., 2026, Nie et al., 2026]. $\square$

Remark 1 (Typed feedback and variance). *Observation 1 is a standard property of zero-variance normalization, recorded here to locate the failure the typed table addresses. Under a group-relative scheme over the alphabets of Definition 1, within-group variance $s_G > 0$ whenever a group contains heterogeneous verdict classes—including the positively scored honest-abstain class $\rho(\text{h-abstain}) = +0.75$ —so the typed table restores exactly the gradient that scalar pass/fail rewards lose when a group saturates. Formalizing this requires the on-policy instantiation of §9; it is a design remark here, not a proved result.*

Observation 1 is the density problem the RLVR fixes target. We keep the scalar where it belongs and open two channels it cannot express:

Definition 2 (Typed reward table). *For reward-shaped stages the feedback is scored by the fixed table $\rho : \mathcal{V}_k \cup \mathcal{V}' \rightarrow \mathbb{R}$,*

$$\rho(v) = \begin{cases} +1.00 & v = \text{cert} \\ -1.00 & v = \text{ref} \\ +0.75 & v = \text{h-abstain (honest decline scored positive)} \\ 0.00 & v = \text{abstain-on-A (decline where decidable)} \\ -1.00 & v = \text{fabricated} \\ -1.00 & v = \text{premise_inv}, \end{cases} \quad (2)$$

where premise_inv is the fidelity-aware term added after the laundering incident ($\lambda_p = 1.00$ registered before retraining).

Definition 3 (Laundering event). *Given task obligation $\text{Ob}(x, y)$ (the prompt’s contract) and artifact-level oracle,*

$$\mathcal{L}(x, y) = \mathbb{I}[\Omega(x, y) = \text{cert} \wedge \neg \text{Ob}(x, y)]. \quad (3)$$

The base reward $\rho \circ \Omega$ is blind to $\mathcal{L}$ by construction: the artifact is sound, so the oracle cannot see the invented precondition. The fidelity-aware reward is $r(x, y) = \rho(\Omega(x, y)) - \lambda_p \mathbb{I}[\text{premise_inv}] - \lambda_f \mathbb{I}[\text{fabricated}]$, and our claim is exactly that $\mathcal{L}$ is measurable—and reducible—only through the second channel.

Definition 4 (Composition objective and gates). *Training is prompt-masked maximum likelihood on oracle-consistent targets with contradicted rows dropped, never relabelled:*

$$\min_{\theta} \mathbb{E}_{k \sim \mathcal{K}} \mathbb{E}_{(x, y^*) \sim \mathcal{D}_k^{\text{filt}}} [-\log \pi_{\theta}(y^* | x)], \quad y^* = \Omega_k\text{-consistent target}, \quad (4)$$

with refresh variants resuming θ on targeted slices of $\mathcal{D}_k^{\text{filt}}$. Evaluation is eight inequalities registered before v2, each a boolean gate $g_j(v) \geq \tau_j$ (direction-aware; thresholds: intent $\tau_1=87$, router $\tau_2=78$, ESC-recall $\tau_3=66$, C-overclaim $\tau_4=5$ (upper), C-recall $\tau_5=90$, A-yield $\tau_6=92$, faults $\tau_7=0$, battery $\tau_8=(6, 5, \text{all})$):

$$\Gamma(v) = \sum_{j=1}^8 \mathbb{I}[g_j(v) \text{ satisfies } \tau_j] \in \{0, \dots, 8\}, \quad v^* = \arg \max_{v \in \text{configs}} \Gamma(v). \quad (5)$$

Remark 2 (How Eq. (2) and Eq. (4) connect). *The reported candidates—including v_4 —are trained by oracle-guided maximum likelihood (Eq. (4)); no policy-gradient update (PPO, REINFORCE, or GRPO) is used anywhere in this paper. The typed table ρ enters the pipeline in exactly two places: (i) as an acceptance filter defining $\mathcal{D}_k^{\text{filt}}$ —rows whose oracle verdict scores negative, or whose laundering detectors (*fabricated*, *premise_inv*) fire, are excluded (dropped, never relabelled); and (ii) inside reward-shaped secondary stages, where ρ scores rollouts directly. RLVF names this feedback interface—typed verdicts from exact oracles flowing into data selection and objectives—not a specific reinforcement-learning algorithm; the group-relative RL instantiation over typed alphabets is posed in §9.*

On the prover axis, the 2×2 matrix specializes the Ω -certifiability pair [Denadai, 2026]:

$$\text{yield}_A = \frac{\#\{\text{claim} \wedge \text{cert on } A\}}{|A|}, \quad \text{overclaim}_C = \frac{\#\{\text{claim on } C\}}{|C|}, \quad \text{recall}_C = \frac{\#\{\text{decline on } C\}}{|C|}, \quad (6)$$

with faults counted separately, never averaged in. The empirical frontier is characterized by the *intent budget function*

$$a^*(c) = \sup\{a(v) : v \text{ configuration, } c(v) \leq c\}, \quad (7)$$

the best intent accuracy attainable under an overclaim budget c ; a^* is weakly increasing by construction, so its *slope* is the exchange rate between the axes. The controlled dose lineage of §4 measures that rate directly: driving Suite-C overclaim $15.2 \rightarrow 0$ moves intent $90.9 \rightarrow 65.5$, i.e. $\Delta a / \Delta c = (90.9 - 65.5) / (15.2 - 0) \approx +1.67$ intent points per overclaim point removed—within the lineage neither endpoint dominates (lower overclaim is purchased with lower intent). Across all nine configurations the budget rises from $a^*(0) = 87.3$ to $a^*(15.2) = 90.9$: at this capacity the two axes trade, and the residual of §4 is the cost of moving along a^* .

3.2 Model, data, and protocol

Model and training. Qwen2.5-Coder-1.5B-Instruct, LoRA rank 8, lr 10^{-5} , batch 1, ~ 600 iterations (~ 1.6 epochs), seed 0, prompt-masked SFT on oracle-derived completions. Refresh variants resume adapters with targeted slices at 3×10^{-6} .

Data. Intent population: parameterized families (unit swaps/drops, entity loss, negation violations, vague asks, ...) re-disposed through the audit oracle; cases whose oracle behaviour contradicts their intended family are *dropped, not relabelled* (v4: 171 population $\rightarrow$ 9 dropped $\rightarrow$ 106 train / 55 eval). Router population: closed-vocabulary synthetic cells plus 740 declared benchmark rows, labelled by the policy (v4: 853 population; ROUTE 517 / ABSTAIN 194 / ESCALATE 142; 106 train / 295 eval). Prover: held-out task suite, $K = 4$ rollouts. Splits are content-hashed and seed-stable; per-component caps, minority oversampling (ESCALATE $\times 5$), plain-prompt curriculum rows, and system-prompt randomization are the recipe knobs. The serving battery’s 26 prompts per cell are excluded from training at both the templated and plain levels.

The eight gates (registered before v2). Templated axis: intent accuracy $\geq 87\%$; router accuracy $\geq 78\%$; ESCALATE recall $\geq 66\%$; undecidable-suite overclaim $\leq 5\%$ with recall $\geq 90\%$; decidable-suite yield $\geq 92\%$; plus 0 faults. Serving axis (a four-cell battery of 26 plain prompts per cell—no task templates): trained+active intent $\geq 6/8$, router $\geq 5/8$, and *every* out-of-fragment prover ask declined. Controls: the base model with only a system prompt must still show its raw prior (10/10 fenced code), and the bare base must show degenerate routing.

Table 1: Nine configurations against the eight gates (best in column bold). C: undecidable-suite overclaim/recall; batt: battery intent/router/prover-declines of 8/8/10.

version	intent	router	ESC	C ovr/rec	A yield	batt	gates
base	54.5	21.7	0.0	79.5/4.5	42.2	—	0
v2	87.3	74.5	66.1	0/100	92.2	3/2/0	4
v3	89.1	76.6	0.0	23.5/76.5	98.9	6/4/5	4
v4	90.9	81.0	84.7	15.2/84.8	97.8	6/6/6	7
v5	58.2	85.4	67.9	0/100	93.3	6/5/6	6
v6	65.5	85.4	67.9	0/100	93.3	6/5/6	6
v6ck900	85.5	78.6	57.6	12.9/87.1	96.1	6/3/6	4
v7	83.6	84.1	56.9	1.5/98.5	92.2	8/5/7	6
v8	87.3	58.0	94.9	68.9/6.1	100	8/5/0	5
soup	80.0	87.1	84.7	9.8/90.2	94.4	6/4/7	5
cosine	87.3	70.8	0.0	26.5/65.9	93.9	6/2/5	4

Evaluation instrument. The prover axis is scored on a frozen suite of 184 tasks (0 dropped, 0 oracle mismatches) in the standard 2×2 matrix: Suite *A* ($n = 120$: straight-line code, loop summaries, premises, bounds, linear systems) and Suite *C* ($n = 64$: mutation, subscript, method, membership, identity, type, unbounded loops, premise gaps, non-polynomial expressions—the family that makes overclaim measurable). Labels are machine-validated through the authority at build time, and the manifest pins split hashes and the authority fingerprint (re-pin = re-dispose; all recorded runs re-disposed by byte-identical across one soundness fix). Suite *A* is authority-shaped by construction—we use it to rank configurations against each other, never as an open-world capability claim. The battery’s four cells (trained+active, trained+inactive, base+prompt, base) run sequentially with one process each at temperature 0; the activation system prompt is the original contract prompt verbatim, one prompt serving all three components (the prompt-design test that fixed this is in Appendix A).

Reward design (for reward-shaped stages). The scoring is exactly the typed table ρ of Eq. (2) (certified +1.0, refuted −1.0, honest abstain +0.75, abstain-on-decidable 0), with the two exact-penalty channels added after the laundering incident: *fabricated* (claim names a quantity the source lacks) and *premise_invention* at −1.00 each, i.e. $\lambda_f = \lambda_p = 1.00$ in the fidelity-aware reward of §3.1. A non-vacuity gate counts a certification only with at least one non-premise discharged obligation. Verdict text never enters training as repair instructions.

4 Results

Composition happens, for free only in the large. Every trained version beats base by a wide margin on all three axes (Table 1): the skills do compose into one prior. What does not come for free is calibration *sub-classes*: the primary optimization trade-offs emerge between overclaim, abstention, ESCALATE, and ASK behaviors, rather than across component domains.

The official candidate. v_4 attains $\Gamma(v_4) = 7$ in Eq. (5) (no configuration reaches 8): intent 90.9 ($n=55$, Wilson [80.4, 96.1]), router 81.0 ($n=295$), ESCALATE 84.7 ($n=59$), *A*-yield 97.8 (false-abstain 1.1), 0 faults, battery 6/6/6 with every out-of-fragment ask declined. Sole residual: the held-out C-gap family (overclaim_C = 15.2% > τ_4 , recall 84.8—still $19\times$ better than base on C). Alternate: v7 if C-calibration outranks two sub-bar templated numbers (*C* 1.5/98.5, battery intent 8/8).

Plain-prompt serving is a separable skill. Plain-ask inference is trained explicitly from v3 on—curriculum rows plus system-prompt randomization—and every later version holds it (the pre-v3 record is part of the Appendix battery). Deactivation fails closed: the trained+inactive cell declines 10/10 with zero fences, while the base cells show the raw prior (10/10 fenced, 5/5 fenced+prose). Contract form never leaked: 0 fenced, 0 prose outputs in every trained active cell.

Table 2 gives the battery in full. Controls held in *every* battery: base+prompt shows 10/10 fenced code (the system prompt alone does not teach decline), the bare base shows 5 fenced + 5 prose, and base router is the degenerate always-ABSTAIN (2/8, correct by luck on abstain cells only).

Table 2: Serving battery, trained+active cell (26 plain prompts: 10 prover / 8 intent / 8 router). Prover shapes: abstain/bare/fenced/prose. Every trained cell keeps contract form; the per-version anomalies are catalogued in Appendix A.

version	prover (ab/ba/fe/pr)	intent	router	OOF declines
v2	0/10/0/0	3/8	2/8	—
v3	5/5/0/0	6/8	4/8	all
v4	6/4/0/0	6/8	6/8	all
v6	7/3/0/0	6/8	5/8	all
v7	7/3/0/0	8/8	5/8	all
v8	0/10/0/0	8/8	5/8	none
soup	7/3/0/0	6/8	4/8	all
cosine	5/5/0/0	6/8	2/8	all
base+prompt	0/0/10/0	3/8	2/8	—
base	0/0/5/5	6/8	2/8	—

The frontier. Nine probes map the intent budget function $a^*(c)$ of Eq. (7). The controlled dose lineage ($0 \rightarrow 32$ gap-decline rows) measures the exchange rate directly: Suite- C overclaim $15.2 \rightarrow 0$ with intent $90.9 \rightarrow 65.5$, i.e. $\Delta a / \Delta c \approx +1.67$ intent points per overclaim point removed, so within the lineage no endpoint dominates the other—lower overclaim is purchased with lower intent. Across all nine configurations, a^* rises from 87.3 at budget 0 to 90.9 at budget 15.2. What each configuration contributed to that map—refreshes, schedule, soup—is catalogued class by class in Appendix A: every knob moves *along* the surface rather than off it, and a regression on any gate is a signal to stop, not to average. The structural claim: at 1.5B / rank 8 / ~ 1.6 epochs the residual is a capacity-point property of the surface, not an artifact of the training recipe; the one untested combination is more epochs so the gap family converges before a gentle tail.

Protocol. Every configuration is evaluated with greedy decoding, per-class Wilson intervals, and content-hashed eval splits the model never trained on; the prover axis is 78 held-out tasks $\times K = 4$ (312 rollouts) per version. A secondary DPO arm on mined failures plus mechanical guards is recorded in Appendix A.

5 Reward laundering, and the fix is in the reward

A premise-fidelity probe—registered with the battery before retraining—quantified a channel that the artifact-level oracle of Eq. (3) cannot see by construction: certified outputs whose emitted assertions strengthen an *undeclared* precondition, converting honest failure into certified success.

Schematic example of the class. The obligation is universal—“return the sum of the positive elements, *for any list* xs .” The emitted artifact computes that sum but first asserts `assert len(xs) > 0` as an additional premise. The artifact-level kernel then verifies a *strengthened* claim: the conditional statement ($xs \neq [] \Rightarrow \text{correct}$) under an assumption the obligation never granted. The empty-list case—part of the actual contract—was never proved, yet the artifact certifies. The prompt declares no non-emptiness parameter; the artifact invents one.

Detection. `premise_invention` is detected by an exact, parameter-only, directional check: an emitted assertion whose antecedent quantifies over a parameter the obligation never declares is flagged. No syntactic parser over arbitrary code and no auxiliary model is involved—the check reads the obligation’s declared parameters and the assertion’s antecedent, nothing else.

Measured on the original corrective run, the channel carried 7 of 52 certified outputs (13.5%); the probe and its repair were registered as one prediction, and confirmed:

	A yield (%)	A ovr (%)	C ovr (%)	C recall (%)	inventions
original corrective run	95.6	4.4	26.5	67.6	7
fidelity-aware ($\lambda_p=1.00$)	93.3	4.4	0.0	100.0	0

Table 3: Quantization parity: fp16 reference vs compressed artifacts, identical eight-gate protocol re-run against the fp16 baseline. Decode is tokens/s measured with the MLX runtime on Apple silicon, batch size 1, on a fixed coding prompt (fp16: 55.5); sizes are on-disk artifact sizes.

artifact	config	size	decode	gates	verdict
sft-all13v4 (fp16)	reference	2.9 GB	55.5	7/8	scoring reference
fp8	8-bit float, 8.25 b/w	1.5 GB	~ 100 ($1.8\times$)	7/8 parity	deploy default
q4g64	4-bit group-64	839 MB	151–162 ($2.9\times$)	6/8	max-speed option
m34	mixed 3/4-bit	679 MB	~ 140 ($2.5\times$)	4/8	rejected (3-bit cliff)

The gain generalized beyond the gap family to the whole undecidable side, at a cost of 2.2% decidable-side false abstention. The incident record—what an invention looks like, why the honest-failure controls never produce one—is kept with the battery (Appendix A). The standing lesson: a sound oracle is not sufficient; the reward must see the task-level obligation. The reward, not the model, is the trust boundary.

6 Deployment: quantization measured, not assumed

Table 3: 8-bit reproduces every router class and the battery bit-for-bit; 4-bit’s single slip is the templated ROUTE class ($94 \rightarrow 69.5$) while Suite *C* improves (6/20 gap declines vs 2/20) and the plain-prompt battery remains identical; 3-bit loses Suite *A*-yield -25 points and ASK $\rightarrow 0$. The 4-bit asymmetry—templated classification degrading while decline behavior survives—is reported as observed; we do not have a mechanistic attribution for it (open challenge six, §9). Reading: compression tracks the training frontier—declines survive compression, capacity does not—so deployment moves only on parity evidence: fp8 by default, int4 when throughput outranks the ROUTE gate, fp16 as the scoring reference.

7 Limitations

Training-seed variance is uncharacterized: every reported version uses a single training seed ($S = 1$), and multi-seed replication ($S \geq 3$, the standard publication bar) is the binding open item (§9). Uncertainty we *can* bound is reported as such: evaluation-split uncertainty via Wilson score intervals (intent $n = 55$, ASK cell $n = 9$, ESCALATE $n = 59$), $K=4$ rollouts. One backbone, one capacity point; the budget-function claim is scoped there. The prover suite mirrors the authority’s vocabulary (authority-shaped by construction, disclosed). The battery is $n=8$ per capability—a serving smoke test with control cells, not a statistical benchmark. Cross-backbone results elsewhere suggest capacity transfers while calibration judgment does not—unexplained at $n=3$.

8 Conclusion

Three oracle-derived skills compose in one 1.5B model into a fail-closed prior—measurably, against gates registered before training: 7/8 for the official candidate against the base’s 0/8, intent 90.9, router 81.0, yield 97.8, serving behaviour that fails closed when deactivated (10/10 declines), and a quantized deployment at full parity ($1.8\times$ decode). The residual is quantified by the intent budget function $a^*(c)$: nine probes agree on one exchange rate at this capacity, and every per-configuration diagnosis is catalogued in the appendix. A pre-registered reward-honesty probe found a laundering channel an artifact-level oracle structurally cannot see, and its repair was confirmed on every axis ($7 \rightarrow 0$ inventions, overclaim $26.5 \rightarrow 0.0\%$, recall $\rightarrow 100\%$)—the reward must see the task-level obligation, and now the reward does. Exact oracles everywhere, registered gates, typed verdict reports, and optimization weights treated as secondary to the measured exchange surface: this is the methodology that makes the reported numbers interpretable.

9 Future Work and Open Challenges

This work was conducted within the hybrid-neurosymbolic research program of **Optimality.ai** (neural proposers, symbolic authorities, fail-closed contracts). The open challenges are as follows.

First and binding: replicate the official candidate at $S \geq 3$ training seeds—training-seed variance is currently uncharacterized, and this is the precondition for treating the reported frontier as more than a single-seed characterization. Second, run the one untested training knob (more epochs, followed by a gentle decay) to determine whether the Suite- C residual is a capacity property or a convergence artifact. Third, instantiate the group-relative RL arm over typed verdict alphabets: GRPO with reward ρ (Eq. (2)) rather than a scalar, so the group signal carries honesty (abstain vs overclaim), the variance argument of Remark 1 made empirical, and the feedback interface of Remark 2 exercised as reinforcement learning rather than filtering. Fourth, scale capacity to ask whether the budget function $a^*(c)$ (Eq. (7)) shifts or persists. Fifth, replicate the judgment-transfer question across backbones under a controlled factorial design until the $n = 3$ confound dissolves. Sixth, attribute the 4-bit templated-ROUTE degradation (§6) mechanistically—class-specific quantization sensitivity is observed, not explained.

A Battery of Tests

The main text presents the primary empirical results; this appendix details the underlying validation and diagnostic experiments—every registered probe, control, and per-configuration diagnosis.

A.1 Per-configuration signatures

Each row was diagnosed from the class-level ledger before any further knob was turned (Table 4).

Table 4: Per-configuration signatures across the nine probes.

version	signature
v3	router ESCALATE $\rightarrow$ 0%: minority swamped by ABSTAIN oversampling
v5	intent blocked $\rightarrow$ 37.5% (answers ASK); A false-abstain 1.1 $\rightarrow$ 12.8—one-sided decline demos taught a surface heuristic
v6	staggered convergence: intent peaks at ck800–900 (85.5), collapses to 65.5 at ck1000 while gap/C and router finish
v7	intent-only refresh lifts intent +18.1 but erodes unshown ESCALATE 67.9 $\rightarrow$ 56.9
v8	refresh composes exactly what it shows: ESC 56.9 $\rightarrow$ 94.9, but ABSTAIN 88.7 $\rightarrow$ 25.8 and gap attempts 20/20 $\rightarrow$ 0/20; serving: 0/10 declines (twin-heavy slice, no out-of-fragment asks in the slice)
soup	composes router/ESC/C but ASK collapses to 11% (subspace interference, not midpoint)
cosine	intent remains stable (87.3), though late training iterations damage previously converged classes; gap/ESC never finish learning—the same steps carry damage and progress

A.2 Serving-battery and prompt-bleed tests

(i) v2 reached C 0/100 and A 92.2 yet scored 0 prover declines and 3/8 intent on plain prompts: templated calibration does not transfer to plain asks untrained—the test that forced the v3 curriculum. (ii) v8’s twin-heavy refresh placed no out-of-fragment ask in its slice: 0/10 declines, OOF column not exercised. (iii) A draft activation prompt that added per-component role paragraphs was measured to bleed answers across skills and rejected; the shipping prompt is the original contract prompt verbatim, one prompt serving all three. All three sit beside the control rows of Table 2.

A.3 Reward-laundering incident record

The channel of §5 in full: an invention is an emitted assertion that strengthens an undeclared precondition—e.g. promoting a contextual assumption into a checked premise—so the artifact verifies while the task obligation is violated. Rate on the original corrective run: $\hat{\mathbb{E}}[\mathcal{L}] = 7/52 = 13.5\%$. Controls: base and self-imitation policies never invent ($\hat{\mathbb{E}}[\mathcal{L}] = 0$); they fail honestly, which is why the channel is attributable to the reward shape rather than the data. Repair as registered: $\lambda_p = 1.00$ (params-only, directional check), gap family enriched 3 $\rightarrow$ 8, retrain; outcome in the table of §5: 0 inventions, C -overclaim 26.5 $\rightarrow$ 0.0%, C -recall 67.6 $\rightarrow$ 100.0%, A -yield cost 2.2% false abstention.

A.4 Secondary arm: DPO

A DPO arm on mined v4 failures plus mechanical guards ($\beta = 0.1$, ~ 620 pairs) reached intent 85.5, router 82.7, ESCALATE 84.7, C 18.2/81.8, A -yield 98.9, 0 faults, battery 6/6/6— $\Gamma = 7$ at best, missing the same two gates as the SFT frontier, consistent with the surface being a capacity property rather than an optimizer property.

A.5 Registered tests and outcomes

Table 5: The registration ledger: every pre-registered test, the claim it challenged, and its outcome.

test	challenge	outcome
eight gates (τ_j fixed before v2)	whether composition beats base under pre-commitment	$\Gamma(v_4) = 7$ vs base 0 — pass
serving battery controls	whether the system prompt alone teaches decline	base+prompt 10/10 fenced (control holds)
activation-prompt role paragraphs	whether prompts must stay one contract	bleed measured; draft rejected
reward-fidelity probe (λ_p)	whether an artifact-level oracle suffices	channel found (7/52), repair confirmed (0)
dose probe on gap-decline rows	whether a^* is tradeable	monotone: 15.2 $\rightarrow$ 0 over-claim costs 90.9 $\rightarrow$ 65.5 intent
quantization parity (re-run gates)	whether compression preserves behaviour	fp8 7/8 parity, battery bit-for-bit

References

- Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.
- C. K. Chow. An optimum character recognition system using decision functions. *IRE Transactions on Electronic Computers*, EC-6(4):247–254, 1957.
- Karl Cobbe, Vineet Kosaraju, Yura Burda, et al. Training verifiers to solve math word problems. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pages 12359–12375, 2021.
- Eduardo Perez Denadai. Ω -certifiability: Claiming exactly what an oracle can certify. *Optimality.ai Technical Report*, 2026. Companion paper.
- Tim Dettmers, Mike Lewis, Younes Belkhalak, et al. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems 35*, 2022.
- Elias Frantar, Saleh Ashkboos, Torsten Hoefer, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.
- Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- Edward J. Hu, Yelong Shen, Phillip Wallis, et al. LoRA: Low-rank adaptation of large language models. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, 2022.
- Gabriel Ilharco, Mitchell Wortsman, Ross Pinto, et al. Editing models with task arithmetic. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.
- Xue Jiang, Yihong Dong, Mengyang Liu, et al. CodeRL+: Improving code generation via reinforcement with execution semantics alignment. *arXiv preprint arXiv:2510.18471*, 2025.

- James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, et al. Overcoming catastrophic forgetting in neural networks. In *Proceedings of the National Academy of Sciences*, volume 114, pages 3521–3526, 2017.
- Harrison Lee, Hubert Pham, Maggie Lao, et al. RLAIIF: Scaling reinforcement learning from human feedback with AI feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 10509–10524, 2023.
- Ziheng Li, Liu Kang, Feng Xiao, et al. Outcome-grounded advantage reshaping for fine-grained credit assignment in mathematical reasoning. *arXiv preprint arXiv:2601.07408*, 2026.
- Hunter Lightman, Vineet Kosaraju, Yura Burda, et al. Let’s verify step by step. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.
- Ji Lin, Jiaming Tang, Haotian Tang, et al. AWQ: Activation-aware weight quantization for large language model compression and acceleration. In *Proceedings of the 1st Workshop on Efficient Systems for Foundation Models (ES-FM)*, 2024.
- Zichen Liu, Changyu Chen, Wenjun Li, et al. Understanding R1-zero-like training: A critical perspective. *Proceedings of the Conference on Language Modeling (COLM)*, 2025.
- David Lopez-Paz and Marc’Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems 30*, pages 6467–6476, 2017.
- Amritansh Mishra, Supriyo Chakraborty, and Berkcan Kapusuzoglu. On the policy gradient foundations of group relative policy optimization: Credit assignment, gradient sparsity, and rank collapse. *arXiv preprint arXiv:2606.29238*, 2026.
- Wenhua Nie, Jianan Wu, Junlin Liu, et al. Gradient starvation in binary-reward GRPO: Why group-mean centering fails and why the simplest fix works. *arXiv preprint arXiv:2605.07689*, 2026.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward hacking. *arXiv preprint arXiv:2212.06305*, 2023.
- Karn Tiwari, Varnith Chordia, and A. P. Prathosh. DiffGate: Difficulty-gated teacher guidance for on-policy distillation. *arXiv preprint arXiv:2610.04596*, 2026.
- Jonathan Uesato, Nate Kumar, Csaba Szepesvári, et al. Solving math word problems with process- and outcome-based feedback. In *Advances in Neural Information Processing Systems 35*, 2022.
- Jingyi Wang, Lei Zhu, Tengjin Weng, et al. GRPO-VPS: Enhancing group relative policy optimization with verifiable process supervision for effective reasoning. *arXiv preprint arXiv:2604.20659*, 2026.
- Mitchell Wortsman, Gabriel Ilharco, Samir Gadre, et al. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pages 23965–23998, 2022.
- Qiyang Yu, Zhengyuan Zhang, Ruofeng Zhu, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, et al. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36, Datasets and Benchmarks Track*, 2023.

Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The abstract states the 7/8-gate result, the mapped frontier, and the single-seed caveat up front.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: A dedicated limitations section covers single seed, eval sizes, authority-shaped suite, one capacity point, and battery scope.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: Observation 1 (vanishing group signal) is stated with assumptions ($s_G = 0$, output-only reward) and derived from the standard zero-variance property, cited to prior analyses; Remark 1 marks the typed-variance extension as design commentary, not a theorem; all other numbered equations are definitions or objectives.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Gates registered before training, manifests with split hashes, authority fingerprints, full version table, battery transcripts, and reproduce commands are disclosed.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- Reproducibility requires a reasonable avenue for the reported results, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Adapters, ledgers, and harness will accompany the final version; the paper discloses gate tables, manifests, and battery matrices sufficient to audit results.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: Sections 3-4 specify model, LoRA config, data construction, drops, splits, knobs, and evaluation.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [\[Yes\]](#)

Justification: Wilson 95

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).

- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [\[Yes\]](#)

Justification: Each run is 600 LoRA iterations on a 1.5B model (laptop-class, MLX); quantization parity runs re-use the same harness; costs are modest and described.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the code of ethics appropriate to the authors' field and institutions?

Answer: [\[Yes\]](#)

Justification: The authors have reviewed the applicable code of ethics.

Guidelines:

- The answer NA means that the authors have not reviewed a code of ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [\[Yes\]](#)

Justification: Fail-closed decline behavior and reward-laundering analysis inform safer deployment; failure modes and their fixes are reported.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Small research adapters of an Apache-licensed base model with evaluation gates; no special safeguards beyond the fail-closed defaults described.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Base model license (Apache-2.0 family) and benchmark provenance will be listed with the release.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: Ledgers, batteries, dataset manifests, and the harness are new assets documented for release.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: No crowdsourcing.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- Under prevailing codes of ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: No human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the applicable code of ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The 1.5B LLM is the object of study; training data are oracle-derived completions and the procedure is fully specified.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.