Mixed-Harness Reinforcement Learning

Supplementary Methods, Results, and Evidence

Companion to *How Can Coding-Agent RL Generalize Across Harnesses?*. Updated October 7, 2026.

This document explains how the report's comparisons were constructed, provides the detailed results behind the figures, and identifies the limits of the available evidence. It describes the current five-model trajectory comparison; earlier two- and four-model subsets are superseded for the response-length figure.

S1. Reading the results

A **training strategy** describes how a checkpoint was produced; an **evaluation harness** is the software interface through which it solves a task. The **instruction-tuned baseline** is Qwen3.5-35B-A3B, evaluated without the additional reinforcement learning (RL) studied here. **Mixed-harness reinforcement learning (MH-RL)** produces one shared checkpoint using OpenHands SDK, Claude Code, and OpenCode. The single-harness RL models are **OH-RL**, **CC-RL**, and **OC-RL**, respectively. **Equal-weight parameter averaging (AVG)** averages the single-harness model checkpoints; **task-vector summation (SUM)** adds their updates to the instruction-tuned baseline. These labels match the main article and all figures.

Accuracy is the percentage of benchmark tasks resolved. A dash denotes an unavailable result, not a zero. The generation-only length is cumulative model output across the complete agent trajectory, including intermediate reasoning and tool-call generation. It excludes prompt and tool-return tokens and is not merely the final answer's length. the response-length figure reports response length (including saved tool replies, estimated), tokens per turn (model generation only), and turns in three panels. Generation-only cumulative totals remain available in this supplement as an additional diagnostic. Section S7 gives the definitions and qualifications.

The curated accuracy matrix and the trajectory-length comparison use different source collections. Archived trajectories need not reproduce every entry in the latest curated accuracy matrix. Neither a matching score nor a common harness name establishes identical serving or execution settings.

S2. Complete evaluation results

The following tables preserve the complete curated cross-harness score matrix. The main article compares MH-RL with the corresponding single-harness model for each evaluation harness in Figure 1. Supplementary Table S1 summarizes benchmark averages and checkpoint counts; the remaining tables retain the complete cross-harness results. The latest author-supplied results are authoritative except for SUM, which is retained from the historical Eval Studio reference. Values are percentages.

Deployment comparison

Table S1. Mean resolved-task percentages with one shared checkpoint or one model per harness (three checkpoints). Harnesses are weighted equally; bold and underlining mark the highest and second-highest values.

Deployment strategy	Checkpoints required	Mean resolved tasks (%)		
		SWE-bench Verified	SWE-bench Pro	SWE-bench Multilingual
Instruction-tuned baseline	1	61.20	37.42	49.90
One model per harness	3	<u>68.40</u>	**40.91**	<u>55.65</u>
MH-RL	1	**68.67**	<u>40.90</u>	**55.66**

The single-harness deployment strategy assigns OH-RL to OpenHands SDK, CC-RL to Claude Code, and OC-RL to OpenCode, requiring three distinct checkpoints in total and one model per evaluation. MH-RL and the instruction-tuned baseline each reuse one checkpoint across all three harnesses. Checkpoint counts describe deployment requirements; training compute is not matched.

S2. SWE-bench Verified

Training strategy	OpenHands SDK	Claude Code	OpenCode
Instruction-tuned baseline	64	62.4	57.2
OH-RL	70.4	67.4	66.2
CC-RL	67.4	68.2	68.6
OC-RL	67.8	66.8	66.6
MH-RL	70.2	67.4	68.4
AVG	67.4	66.8	65
SUM	70.2	67	69.8

S2. SWE-bench Pro

Training strategy	OpenHands SDK	Claude Code	OpenCode
Instruction-tuned baseline	39.8	34.98	37.48
OH-RL	42.3	39.75	40.63
CC-RL	44.05	37.76	43.09
OC-RL	43.6	38.3	42.68
MH-RL	40.9	40.9	40.9
AVG	41.59	38.58	42.68
SUM	43.5	43.91	45.01

S2. SWE-bench Multilingual

Training strategy	OpenHands SDK	Claude Code	OpenCode
Instruction-tuned baseline	51.7	47.33	50.67
OH-RL	55.33	52.33	53.67
CC-RL	59.33	54.3	52.67
OC-RL	57.3	53.67	57.33
MH-RL	56	53.3	57.67
AVG	56	51	57
SUM	58	53	58.33

Score coverage and runtime qualifications

The curated SWE-bench Verified matrix uses the column labeled for a 4,800-second agent budget. Longer-budget columns are excluded. The latest supplied SWE-bench Pro and SWE-bench Multilingual results do not specify every runtime budget, so configuration matching across all cells has not been established.

Two historical SUM entries warrant specific qualifications. SWE-bench Pro through OpenHands SDK (43.50%) comes from the curated reference table; its raw run was not independently located. SWE-bench Multilingual through OpenCode (58.33%) comes from the job summary, which reports 175 successes out of 300. An earlier audit could read only 299 trial files, with 174 successes. The report retains the job summary and records this discrepancy.

Benchmark averages and harness sensitivity

A benchmark mean equally weights its three harnesses. The single-harness deployment strategy selects OH-RL through OpenHands SDK, CC-RL through Claude Code, and OC-RL through OpenCode. It does not select the best score in each column.

Benchmark	Single-harness mean	MH-RL mean
SWE-bench Verified	68.40%	68.67%
SWE-bench Pro	40.91%	40.90%
SWE-bench Multilingual	55.65%	55.66%

The largest difference between these benchmark means is 0.27 percentage points. Across the nine individual benchmark–harness pairs, MH-RL's largest shortfall relative to the corresponding single-harness model is 1.78 points, on SWE-bench Pro through OpenCode. These are descriptive results, without an equivalence test or confidence interval.

Harness sensitivity is the maximum minus minimum score within a benchmark. The report averages those three spreads for each training strategy; this quantity is not variance across seeds.

Training strategy	SWE-bench Verified	SWE-bench Pro	SWE-bench Multilingual	Mean spread
OH-RL	4.20	2.55	3.00	3.25
CC-RL	1.20	6.29	6.66	4.72
OC-RL	1.20	5.30	3.66	3.39
MH-RL	2.80	0.00	4.37	2.39

All spreads are percentage points. A spread across all harnesses is different from a native-to-transfer drop; for example, CC-RL's 6.66-point range on SWE-bench Multilingual is not its native-to-OpenCode drop of 1.63 points.

S3. Training settings and model merging

The exported configurations identify Qwen3.5-35B-A3B as the common starting model and OpenSWE data configurations labeled for 2,699 tasks. Policy updates use VeOmni; rollouts use asynchronous vLLM. All four runs use GSPO with GRPO advantage estimation, AdamW, a constant learning rate of 0.000001, and a policy-update mini-batch size of 64.

Setting	MH-RL	OH-RL	CC-RL	OC-RL
Maximum rollout context, tokens	160,000	200,000	200,000	200,000
Response limit, tokens	130,000	170,000	170,000	170,000
Exported trainer node count	4	2	2	2
Configured total epochs	3	3	5	3
Policy loss	GSPO	GSPO	GSPO	GSPO

These are exported settings, not hardware counts inferred from run names. All four exports include VeOmni router replay mode R3. Training-data paths, validation-data paths, and trajectory filters differ; equal task membership, effective harness proportions, and total training compute have not been independently established.

MH-RL, OH-RL, and OC-RL declare context-error, environment-failure, and timeout filters, a minimum response length of eight tokens, and no overlength or turn-cap filtering. The CC-RL export instead declares timeout and environment-setup failure drop reasons. Different schemas and implementation versions limit direct comparisons of filtering behavior.

On SWE-bench Pro, the updated AVG results are 41.59% through OpenHands SDK, 38.58% through Claude Code, and 42.68% through OpenCode. The latter two scores were supplied by the author on October 7, 2026. Their equally weighted mean is 40.95%, compared with 40.90% for MH-RL. The main article's model-merging table retains an OpenHands SDK-only Pro comparison because SUM results through Claude Code and OpenCode remain unavailable. The 0.05-point difference between the three-harness means does not establish statistical superiority.

AVG is the equal-weight average of the three single-harness model checkpoints. SUM equals the base plus the sum of the updates from the base to each single-harness model, each with coefficient one. These are distinct update scales. Neither rule makes the total training compute of three single-harness models equal to that of one MH-RL run.

The evaluated MH-RL checkpoint is step 90. The training curves continue beyond that checkpoint; the end of a plotted curve is not a new benchmark evaluation.

Shared-policy objective and merge rules

Let θ be the shared policy parameters, x a training task drawn from $\mathcal{D}$, and τ the interaction trajectory through harness h. Write $P_{\theta,h}(\tau \mid x)$ for the trajectory distribution induced jointly by the policy, harness, and task environment. Define

$$J_h(\theta) = \mathbb{E}_{x\sim\mathcal{D},\,\tau\sim P_{\theta,h}(\cdot|x)}[R(\tau)].$$

With uniform harness selection at rollout generation, the intended objective is

$$J_{\text{mix}}(\theta) = \frac{1}{3}\left[J_{\text{OH}}(\theta) + J_{\text{CC}}(\theta) + J_{\text{OC}}(\theta)\right].$$

The one-third probability selects a harness; it does not assign one-third probability to a particular trajectory. All accepted trajectories update the same policy. Filtering, rollout duration, and admission can affect realized update proportions, so this expression describes the sampling objective rather than a measured balance in every update.

For model merging, θ_0 is the common starting model and $\Delta_h = \theta_h - \theta_0$ is a single-harness model's learned update. The two rules are

$$\theta_{\mathrm{AVG}} = \theta_0 + \frac{1}{3}\sum_h \Delta_h, \qquad \theta_{\mathrm{SUM}} = \theta_0 + \sum_h \Delta_h.$$

S4. Trajectory-length comparison

Task selection and interpretation

Within each evaluation harness, a task enters the response-length figure only if every displayed model has reward one, no recorded exception, and a numeric exact output-token count. Within each harness, all five models and all three metric panels use the same task subset. Different harnesses can use different subsets.

The result-field source is `agent_result.n_output_tokens`. Counts cover the entire agent trajectory. Rounded token displays in terminal logs are excluded. Each mean below is rounded only for presentation after computation from exact per-task records.

S4. OpenHands SDK

All five models are compared on the same 110 successful tasks.

Training strategy	Mean output tokens	Median output tokens
OH-RL	17,101.0	15,896.5
CC-RL	13,832.8	12,116.5
OC-RL	12,558.3	10,759.5
SUM	24,465.2	24,328.0
MH-RL	12,684.5	11,333.5

MH-RL uses 48.2% fewer output tokens than SUM on this subset.

S4. Claude Code

All five models are compared on the same 203 successful tasks.

Training strategy	Mean output tokens	Median output tokens
OH-RL	13,511.1	10,814.0
CC-RL	15,953.5	13,886.0
OC-RL	13,614.8	10,585.0
SUM	30,456.7	29,219.0
MH-RL	13,276.7	11,373.0

MH-RL uses 56.4% fewer output tokens than SUM on this subset.

S4. OpenCode

All five models are compared on the same 225 successful tasks.

Training strategy	Mean output tokens	Median output tokens
OH-RL	13,220.2	11,447.0
CC-RL	13,246.6	10,398.0
OC-RL	16,098.7	14,623.0
SUM	29,629.1	30,078.0
MH-RL	14,037.1	10,905.0

MH-RL uses 52.6% fewer output tokens than SUM on this subset.

Recovery of training-harness evaluation records

Single-harness model	Training validation step	Unique tasks	Reproduced successes	Attempt records
OH-RL	95	500	352 (70.4%)	521
CC-RL	110	500	341 (68.2%)	540
OC-RL	50	500	333 (66.6%)	505

OH-RL and OC-RL records were recovered from the original training-validation batches. For each task, the reconstruction selects the earliest exception-free attempt; if every attempt has an exception, it retains the last attempt. Selection does not depend on reward or token availability. The resulting success totals reproduce the logged validation scores. This is a reconstruction from trial files, not an original runtime-return manifest.

CC-RL follows its separately audited selection: retain the earliest execution of each Verified task, excluding eight non-Verified records. All 40 later duplicate executions start after every first execution has begun. The selected 500 tasks reproduce the logged step-110 result. Its existing 203-task comparison subset is unchanged.

For OH-RL, 194 of 500 selected tasks retain exact output counts. Of the 352 successful tasks, 141 have exact counts and 211 do not. All 211 missing exact counts coincide with a missing `agent/trajectory.json`, from which the OpenHands reader populates the token field. Their terminal logs retain rounded output displays, but those displays are not substituted. Consequently, the 110-task OpenHands comparison also conditions on artifact availability and may not represent all successful tasks.

For OC-RL, all 333 successful tasks have exact token counts. Six have a recorded agent-timeout exception and are excluded under the common rule, leaving 327 successful exception-free records before intersecting with the other models.

Configuration differences and scope

Archived cross-harness evaluations of the single-harness models use 4,800-second agent limits, as do the SUM and MH-RL evaluations. Recovered native CC-RL has a 2,400-second override and maximum. Native OH-RL and OC-RL record a maximum timeout cap of 3,600 seconds, no explicit override, and an agent timeout multiplier of one; this cap alone does not establish the effective timeout for every task.

Serving configuration also differs. In the historical SUM–MH-RL audit, SUM used tensor parallelism of four, eager execution, concurrency 50, and maximum context 262,144; MH-RL used tensor parallelism of eight, compiled execution, concurrency 96, and maximum context 200,000. These differences remain relevant after adding single-harness baselines.

The recovered archived OC-RL transfer runs resolve 336/500 tasks through OpenHands SDK and 330/500 through Claude Code. Those are the runs behind the token comparison, not replacements for newer curated accuracy scores. SUM through Claude Code combines two disjoint 250-task jobs. MH-RL through Claude Code uses the archived September 19 08:50:53 run.

The comparison is observational and conditional on joint success and exact-record availability. It excludes failures and does not measure full-evaluation token cost, equal-compute efficiency, or intrinsic wall-clock speedup. No new training or evaluation jobs were launched to fill the figure.

S5. Training dynamics and metric processing

the supplementary training-progress figure combines the aggregate training score with a diagnostic of reward within each harness during the same MH-RL run. The aggregate score uses the logged `critic/score/mean` metric. Global training step is taken from `training/global_step` when available, rather than the asynchronous log-row step. Duplicate metric/step pairs retain the last recorded value; missing points are not zero-filled.

The logs store a harness-selection indicator and a reward contribution that is zero when that harness is not selected. For a recorded rollout, let H be the selected harness and R its reward. The average reward for harness h is estimated by normalizing the logged contribution by the selection frequency:

$$\hat{r}_h = \frac{\mathrm{mean}[\mathbf{1}(H = h)R]}{\mathrm{mean}[\mathbf{1}(H = h)]}.$$

Without this normalization, a more frequently sampled harness can have a larger logged reward contribution even when its conditional performance is unchanged.

For harness h, define R_h as the logged reward contribution and I_h as the logged selection indicator. Unselected harnesses receive zero contribution. The conditional reward at a step is the logged mean of R_h divided by the logged mean of I_h. For a multi-step window, divide the sum of logged reward means by the sum of logged selection means; do not take an unweighted average of per-step ratios.

Faint traces show per-step observations; solid curves use trailing ten-step windows. The curves show steps 1–126, with the evaluated MH-RL step 90 marked. The early and checkpoint windows below are steps 1–20 and 71–90. These 20-step windows were chosen for descriptive reporting; they are not predefined training phases. The endpoint of step 90 matches the evaluated checkpoint, whereas the available training record continues to step 126. The exact changes depend on the window choice.

Harness within MH-RL	Early window	Checkpoint window	Change
OpenHands SDK	55.68%	63.62%	+7.94 pp
Claude Code	50.14%	56.31%	+6.16 pp
OpenCode	51.28%	61.75%	+10.47 pp

As a sensitivity check, we repeated the comparison using windows of 5, 10, 20, and 30 steps, comparing each initial window with an equal-width window ending at step 90. All three harnesses showed positive changes for all four window widths, but the magnitudes varied. This check describes sensitivity to window selection; it is not a statistical significance test or an evaluation on a fixed task set.

The aggregate MH-RL training score rises from 51.73% in steps 1–20 to 60.68% in steps 107–126. These are training-rollout statistics, not independent held-out evaluations. Conditional rewards are recorded before downstream filtering, so they may differ from the filtered training score. Changes in task difficulty and execution conditions also remain uncontrolled.

The source keys are `timing_s/agent_loop/harness_reward/{harness}/mean` and `timing_s/agent_loop/harness_selected/{harness}/mean`. Despite their timing prefix, they encode reward contributions and selection indicators, not seconds. The implementation initializes unselected harness rewards to zero and assigns verifier reward to the selected harness. The preserved source snapshot records the selection and reward-assignment logic.

The previously plotted invalid-trajectory ratio measures how many rollouts a run's filtering rules reject. It is an operational diagnostic sensitive to timeouts, environment failures, and filter configuration. It is not used as evidence that mixing itself improves task solving.

S6. Source provenance and reproducibility

The curated score matrix combines the latest author-supplied results with the qualified historical SUM reference. The training plots use exported dashboard metrics and configurations. The trajectory-length analysis uses per-task archived result files; selected task IDs and source paths are retained in the local evidence bundle.

The main article's Training Dashboards section links to the original live curves. The identifiers below document the runs used in the October 5 snapshot; subsequent dashboard updates need not match the report.

- **MH-RL:** `mixed-harness-q35-35b-4n-h20-20260905`.
- **OH-RL:** `openhands-q35-35b-3n-openswe2699-200k-20260717`.
- **CC-RL:** `claudecode-q35-35b-openswe2699-200k-20260718-FULL`.
- **OC-RL:** `opencode-q35-35b-3n-openswe2699-200k-20260728-FULL`.

The retained evidence includes original attempt summaries, selected per-task records, exact common-task intersections, metric exports, figure-generation scripts, and source hashes. A reproduction should preserve the documented task-selection rules, recompute every model on each panel's common intersection, and distinguish the curated accuracy matrix from the trajectory archive.

This supplement presents the current analysis as a coherent document. It does not treat earlier figure versions, editing history, or raw implementation dumps as additional experimental results. Background references for the benchmarks, algorithms, and frameworks appear in the main article's reference list.

S7. Metric definitions and additional token accounting

All measurements below use the same 2,690 model–task records as the previous generation-and-rounds audit: 110 shared successful tasks through OpenHands SDK, 203 through Claude Code, and 225 through OpenCode, with five models per task. No tasks were added or dropped. Existing generated-token means and turn counts are unchanged.

Token accounting

Let G_i be logged generated tokens for task i, O_i the re-tokenized saved tool-reply text, and T_i recorded model turns. We report four distinct quantities:

1. **Generation-only totals (supplement only):** mean(G_i), the mean cumulative model generation per task. Read `agent_result.n_output_tokens`, verified against `trajectory.final_metrics.total_completion_tokens` and the sum of per-turn completion counts where present. It includes CoT insofar as reasoning is retained in completion accounting, as well as tool-call generation and final answers. A missing separate reasoning-text field does not imply zero CoT.
2. **Response length (including saved tool replies; estimate):** mean(G_i + O_i). For every `observation.results[].content` string, encode it with the Qwen3.5 tokenizer and `add_special_tokens=False`; sum once per recorded reply. We add these text-token counts to logged generation. No initial prompt, repeated prompt history, or padding is added. Existing text formatting and metadata embedded in the exported content are retained.
3. **Tokens per turn:** sum(G_i) / sum(T_i), a pooled average across recorded model turns on the selected tasks. This excludes tool replies. It is deliberately not mean(G_i / T_i), which weights every task equally rather than every turn. Both values are available in the CSV.
4. **Turns:** mean(T_i). OpenHands SDK and OpenCode use recorded agent turns. Claude Code uses agent steps carrying a completion-token metric, preventing separate tool-execution records from being

counted again. Multiple tool calls from one model turn do not create additional rounds. Internal transport retries or hidden harness calls need not appear in this count.

Limits of the tool-inclusive estimate

This estimate is based on archived tool-reply text, not the exact token-ID stream used by the model or trainer. Exported text can contain formatting or metadata that differs from model-visible serialization. Chat-template delimiters, other inserted messages, deleted or truncated content, context compaction, and trajectory segmentation are not reconstructed. Accordingly, it is neither an exact reproduction of training `response_length/mean` nor a guaranteed lower or upper bound on that metric.

Some recorded tool-call IDs have no matching observation result. This can reflect errors, incomplete exports, or calls without a retained response; we cannot assume their unrecorded response length is zero. The inclusive column therefore explicitly says “saved tool replies.” We retain the original paired subsets, report coverage below, and do not impute unseen text. Repeated IDs within the saved replies were checked to prevent an unnoticed duplicate-counting issue.

`total_prompt_tokens` is not added to generation: every request can include the entire earlier history, so summing request inputs repeatedly counts past messages and represents a different processing-cost measure.

OpenHands SDK: the three lengths and rounds

Model	Excluding tools: tokens/task	Including saved tools: estimated tokens/task	Generated tokens/turn	Turns/task
OH-RL	17,101.0	62,576.2	234.9	72.8
CC-RL	13,832.8	44,004.8	252.1	54.9
OC-RL	12,558.3	43,415.0	228.5	55.0
SUM	24,465.2	87,056.0	245.0	99.8
MH-RL	12,684.5	46,940.4	255.2	49.7

Claude Code: the three lengths and rounds

Model	Excluding tools: tokens/task	Including saved tools: estimated tokens/task	Generated tokens/turn	Turns/task
OH-RL	13,511.1	67,475.6	257.0	52.6
CC-RL	15,953.5	64,467.2	273.3	58.4
OC-RL	13,614.8	69,352.6	265.3	51.3
SUM	30,456.7	106,287.5	289.5	105.2
MH-RL	13,276.7	70,016.6	268.7	49.4

OpenCode: the three lengths and rounds

Model	Excluding tools: tokens/task	Including saved tools: estimated tokens/task	Generated tokens/turn	Turns/task
OH-RL	13,220.2	36,248.1	219.9	60.1
CC-RL	13,246.6	35,304.5	233.7	56.7
OC-RL	16,098.7	41,651.7	236.7	68.0
SUM	29,629.1	66,560.0	251.0	118.0
MH-RL	14,037.1	39,836.9	260.0	54.0

Saved-reply coverage

“Unmatched calls” counts recorded call IDs without a matching saved reply; the adjacent count gives affected tasks. This flags uncertainty in the inclusive estimate, not a failure of the logged generation metric.

Harness / model	Saved replies	Unmatched calls	Affected tasks
OpenHands SDK / OH-RL	7,981	24	19 / 110
OpenHands SDK / CC-RL	5,987	47	34 / 110
OpenHands SDK / OC-RL	5,979	67	39 / 110
OpenHands SDK / SUM	10,965	18	15 / 110
OpenHands SDK / MH-RL	5,424	42	33 / 110
Claude Code / OH-RL	10,544	0	0 / 203
Claude Code / CC-RL	11,694	0	0 / 203
Claude Code / OC-RL	10,372	0	0 / 203
Claude Code / SUM	21,158	0	0 / 203
Claude Code / MH-RL	9,864	0	0 / 203
OpenCode / OH-RL	13,265	232	121 / 225
OpenCode / CC-RL	12,861	179	105 / 225
OpenCode / OC-RL	15,236	195	99 / 225
OpenCode / SUM	26,373	72	55 / 225
OpenCode / MH-RL	12,584	29	24 / 225

All saved nonempty reply contents were strings, with no unsupported content types or duplicate reply IDs. Local and remote records were encoded with identical tokenizer files (SHA-256 `5f9e4d4901a92b997e463c1f46055088b6cca5ca61a6522d1b9f64c4bb81cb42`). Original generation and turn-count consistency checks remain valid.

For reuse, download the three-panel figure as PDF or the complete numeric table as CSV.