Agentic Coding Environments

Dataset specification

GR-LLM-03 · LLM · Early access · updated 2026-11-02

1 - Overview

Containerized software and tool-use tasks with automated verifiers.

Runnable tasks for training and evaluating coding and tool-use agents: feature work in real open-source repositories, long-horizon engineering projects, terminal tasks, skill-based tasks and multi-service enterprise workflows. Every task ships with its environment, an automated verifier and a reference solution, ready for reinforcement learning.

- **Task format:** Harbor
- **Verification:** Automated
- **Reference solutions:** Included
- **Repo languages:** 5+
- **Modalities:** Code, Docker environments, Tests, Tool calls

2 - Use cases

- Reinforcement learning for coding and tool-use agents
- Agent evaluation beyond public benchmarks
- Regression testing of agent harnesses

3 - Task families

- **Repository tasks:** Feature requests in real open-source repositories (Python, TypeScript, JavaScript, Go, Rust), graded by fail-to-pass and pass-to-pass tests
- **Long-horizon engineering:** Reimplementation and performance work with byte-exact or benchmark-based verifiers
- **Terminal:** Linux command-line and scientific-computing tasks in containers
- **Skills:** Tasks solved with packaged agent skills and checked by output tests
- **Enterprise services:** Workflows across self-hosted GitLab, ownCloud, Plane and Rocket.Chat for fictional companies, scored against checkpoints
- **MCP tool use:** Multi-step tasks over MCP servers with rule-based ground truth

4 - Each task includes

- **Instruction:** Task statement in English (some skill tasks in Chinese)
- **Environment:** Dockerfile and repository snapshot or seeded services
- **Verifier:** Tests, grader script or checkpoint rubric
- **Reference solution:** Patch or script that passes the verifier

5 - Provenance

- **Tasks:** Newly written; not copies of public benchmark tasks
- **Upstream code:** Open-source repositories under their original licenses

- **Model trajectories:** Not included
- **Volume:** Confirmed per order

6 - Data schema

- `task.toml` **(toml):** Task metadata, environment and timeouts.
- `instruction.md` **(markdown):** Task given to the agent.
- `environment/` **(directory):** Dockerfile and task inputs.
- `tests/` **(directory):** Verifier that scores the agent's result.
- `solution/` **(directory):** Reference solution.

7 - Directory layout

```
agentic-coding-environments/  # indicative; final layout confirmed per delivery
`-- tasks/<task_id>/
    |-- task.toml
    |-- instruction.md
    |-- environment/
    |   `-- Dockerfile
    |-- tests/
    |   `-- test.sh
    `-- solution/
        `-- solve.sh
```

8 - Evaluation sample

- **Availability:** On request
- **Size:** On request
- **Formats:** Harbor tasks, Docker
- **Contents:** Runnable tasks from each family with verifiers and reference solutions

9 - License and contact

- **Graviti Commercial Data License:** Commercial training and evaluation rights. Upstream open-source code keeps its own license.
- **Dataset page:** https://www.graviti.com/datasets/agentic-coding-environments/
- **Sales:** sales@graviti.com