

Simulation RL Environments



Part 1 of a series on simulation RL environments. AI agents re-plan container-ship dockings at the Port of Barcelona, built from the port's real 2024 records.

AUTHOR

[Adithya S Kolavi](#)

AFFILIATION

[Hugging Face](#)

PUBLISHED

October 6, 2026

CODE

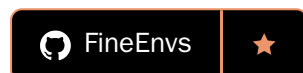


Table of Contents

1 Simulation RL environments

2 PortSimEnv v1

2.1 Try it

3 Why it matters

4 The eval

4.1 Watch a rollout

5 How the reward works

6 Built on real Port of Barcelona data

6.1 From seed data to tasks

6.2 Data sources

6.3 Links

7 Ongoing work

Work in progress

This is an early, experimental series. The environments, numbers and conclusions will change as we go.

Simulation RL environments

Most RL environments for language models are puzzles, code, maths or web tasks. A lot of valuable work looks different. Someone keeps a schedule running under real rules while things keep going wrong. I want agents that can do that work, which raises a general question. How do we build simulation environments, digital twins of real work, that are realistic enough to learn from?

Each part of this series takes one real job, rebuilds it from the records it leaves behind, and grades the agent against an answer that can be proven best.



PortSimEnv v1

PortSimEnv is a deterministic RL environment for berth planning at the Port of Barcelona. It is built from the 1,784 container-ship calls the port recorded in 2024 at its two container terminals, with the real ships, their sizes, berthing times and quay sections. The agent gets one quay, the ships that really called there in a 2024 week, and a week that has just gone wrong. It re-plans the week, deciding for every ship when it docks, where along the quay and with how many cranes.

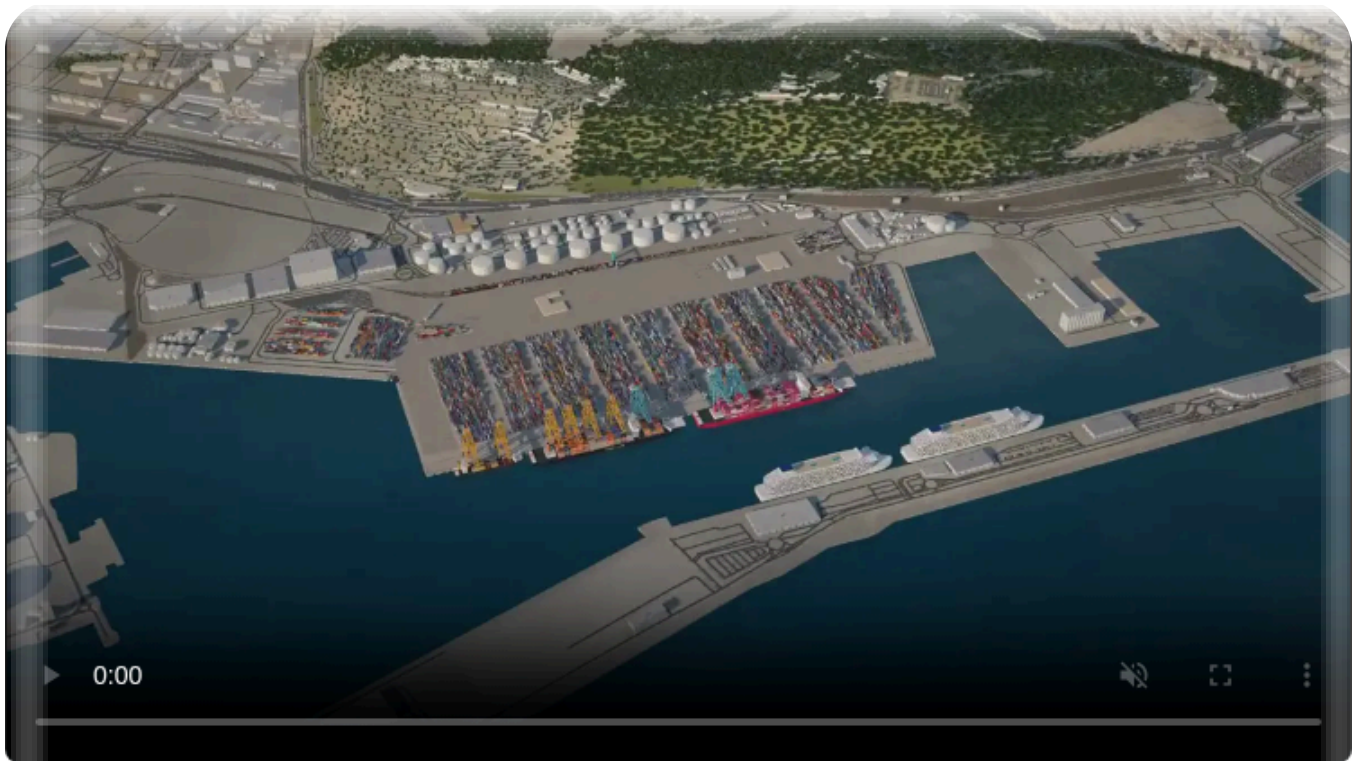
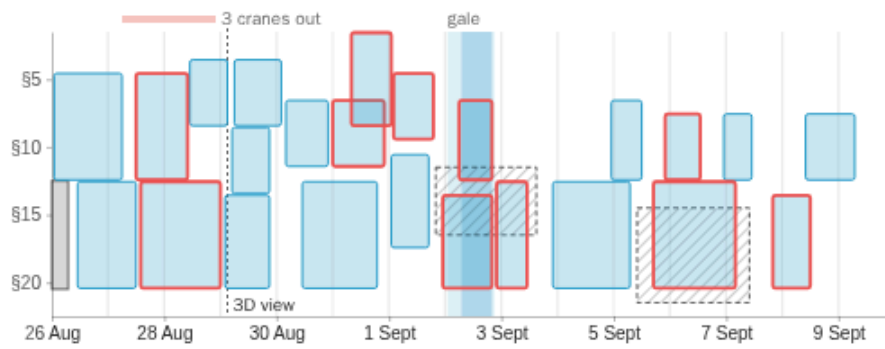


Figure 2 · A recorded eval run, replayed on the 3D twin: GLM-5.3 on week 6 at APM Terminals (17 ships, 9 cranes, two cranes down and two quay closures). It checks three drafts and submits a plan that matches the proven optimum, reward 1.0. The cursor only shows its tool calls. The agent itself sends structured plans.

Before: the port's plan, after the week went wrong

12 rule breaks · reward 0.11

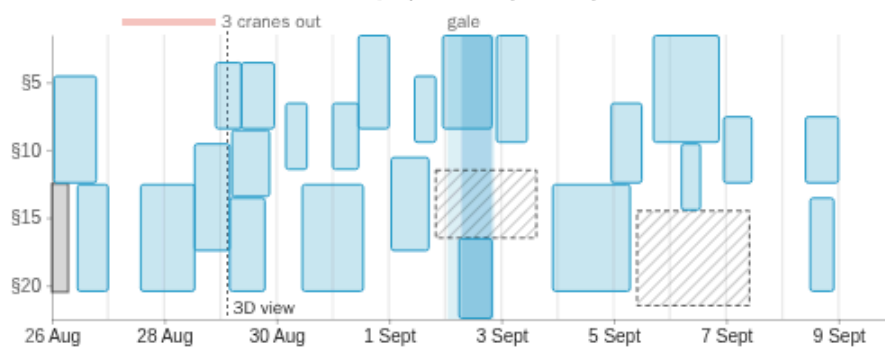


In 3D at Thu 29, 03:00, under the port's plan: this stretch of quay stands empty.

GPT-6.1 Sol's episode `get_situation` → `check_plan` valid, cost 327 → `check_plan` valid, cost 312 → `submit_plan` → reward 1.00

After: GPT-6.1 Sol's re-plan

no rule breaks · reward 1.00 · cost 312, equal to the proven optimum



The same moment under the re-plan: CYPRESS alongside with six cranes, MARTHA A beside it.

ship alongside breaks a rule closed quay gale cranes out

In 12–90 real ships, the quay, its cranes, closures, the weather

Out when, where and with how many cranes each ship docks

Score rules first, then delay against the proven optimum

Every plan plays out on a 3D twin of the real port, built from open data. The agent's berthing hours, quay sections and crane counts turn into ships, tugs and cranes moving through the week.



The harbour: the twin covers the port, Montjuïc and the city, from OpenStreetMap and public terrain.



APM Terminals, storm week 35: GPT-6.1 Sol's plan, which matches the proven optimum. Tugs bring a ship in.



BEST, busy week 6: 25 ships; three worked at once under GPT-6.1 Sol's optimal plan.

The disruptions come from what happens in real ports. Red Sea diversions send ships in late and in bunches, gales stop ships of 300 m or more above 25 knots and every ship above 30, a crane goes down for two days, or an emergency needs a berth now.

Try it

This is the environment's [OpenEnv Space](#), embedded live. Pick a split and a task index, load the episode and plan it with the same three tools the agent gets. The Playground tab is OpenEnv's MCP playground. The code is in [07-simulation-environments/portsim-v1](#) on GitHub.

the PortSimEnv environment, live

Loading the live environment from the Space...

Pick a split and a task index, load **episode**, then plan: drag ships on the chart or type their hour, section and cranes, test with **Check plan** and finish with **Submit plan**.

[Open the environment Space ↗](#)

Figure 5 · It opens on week 7 at APM Terminals (17 ships). Drag ships on the dock chart or type their hour, section and cranes, and the 3D quay follows. Check plan reports rule breaks and cost, 10 times per episode. Submit plan

ends the episode with a single grade.

Why it matters

A ship’s containers only move once it is alongside under a crane, so the berth plan decides how long ships wait, how much fuel they burn at anchor and whether a whole service stays on schedule. Schedules rarely hold, so the plan keeps changing.

up to 9% of a ship's time is spent waiting at anchorage IMO	4.3M TEU Jun 24 Aug 26 of container capacity stuck at anchorages, a record (Aug 2026); 2.5M in June 2024 Maritime Executive / Linerlytica	\$49–72k a day to charter a 4,250–8,500 TEU ship (Oct 2026), before fuel at ~\$930/t Harper Petersen HARPEX
3.5 days average wait for a berth in Barcelona, May 2024 Xeneta	49.9% on time of container ships on time (Aug 2026); late ones 6.8 days late Sea-Intelligence	11% CO₂e of shipping's emissions happen at berth or at anchor; arriving just in time cuts a voyage's fuel by 14% ICCT / IMO

Barcelona alone handled [3.9 million TEU from 2,163 container-ship calls](#) in 2024, and in 2025 it [changed its rules](#) to give berth priority to regular services during congestion. Rotterdam reports that its port-call app [cut waiting by 20%](#).

The eval

Six models each ran the 50 held-out weeks once, with the same three tools, 12 turns and 32k output tokens per turn.

The eval at a glance

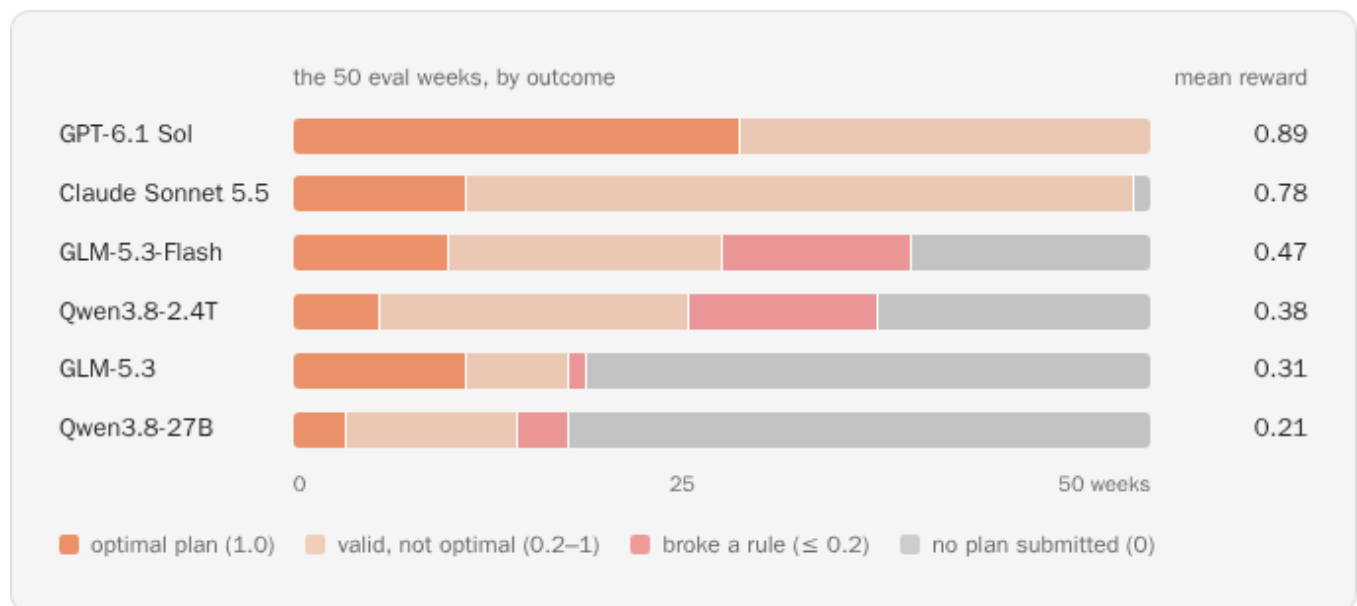


Figure 7 · Each bar is one model's 50 eval weeks by outcome; the number is its mean reward. Hover for counts.

- GPT-6.1 Sol leads at 0.89 and submits a valid plan every week, but it matches the optimum on only 26 of 50, so there is plenty of headroom.
- Every model scores lower on storm and extreme weeks, which stack gales, closures and emergencies.
- The open models lose most of their score before grading. They spend their output budget reasoning and never call `submit_plan`. On the 13 to 25 weeks where they do submit a valid plan, they average 0.75 to 0.97, so getting them to submit is the first thing to train.

Watch a rollout

All 300 eval rollouts replay on the 3D quay in the eval Space, [FineEnvs/PortSimEnv-Eval](#), with the plan at each tool call, the dock chart, the grade and the full transcript.

Rollouts on the 3D quay: five eval weeks, six models

Week Standard · APM · 15 ships Busy · APM · 16 ships Busy · BEST · 25 ships Storm · APM · 24 ships

Extreme · APM · 28 ships

Model GPT-6.1 Sol 1.00 Claude Sonnet 5.5 0.55 GLM-5.3-Flash 0.83 Qwen3.8-2.4T 0.76 GLM-5.3 no plan

Qwen3.8-27B no plan

Loading the live 3D replay from the Space...

Claude Sonnet 5.5: valid, reward 0.55. Cost 381 vs optimum 312 (unavoidable 244) → gap 0.411 → $0.2 + 0.8 \cdot e^{-\text{gap}/0.5}$.

[Open in the eval Space](#) ↗

Figure 8 · Pick a week and a model. The line under the replay explains the reward.

How the reward works

The reward is deterministic, so the same plan always gets the same score. It is computed from the plan alone, without an LLM judge or sampling, once per episode when the agent calls `submit_plan`.



Example · Claude Sonnet 5.5 on the storm week: no rule broken · cost 381 · optimum 312 · unavoidable 244
 $\text{gap} = (381 - 312) / (312 - 244 + 100) = 0.411 \rightarrow \text{reward} = 0.2 + 0.8e^{-0.411/0.5} = 0.55$

Reward against cost, with each model's plan

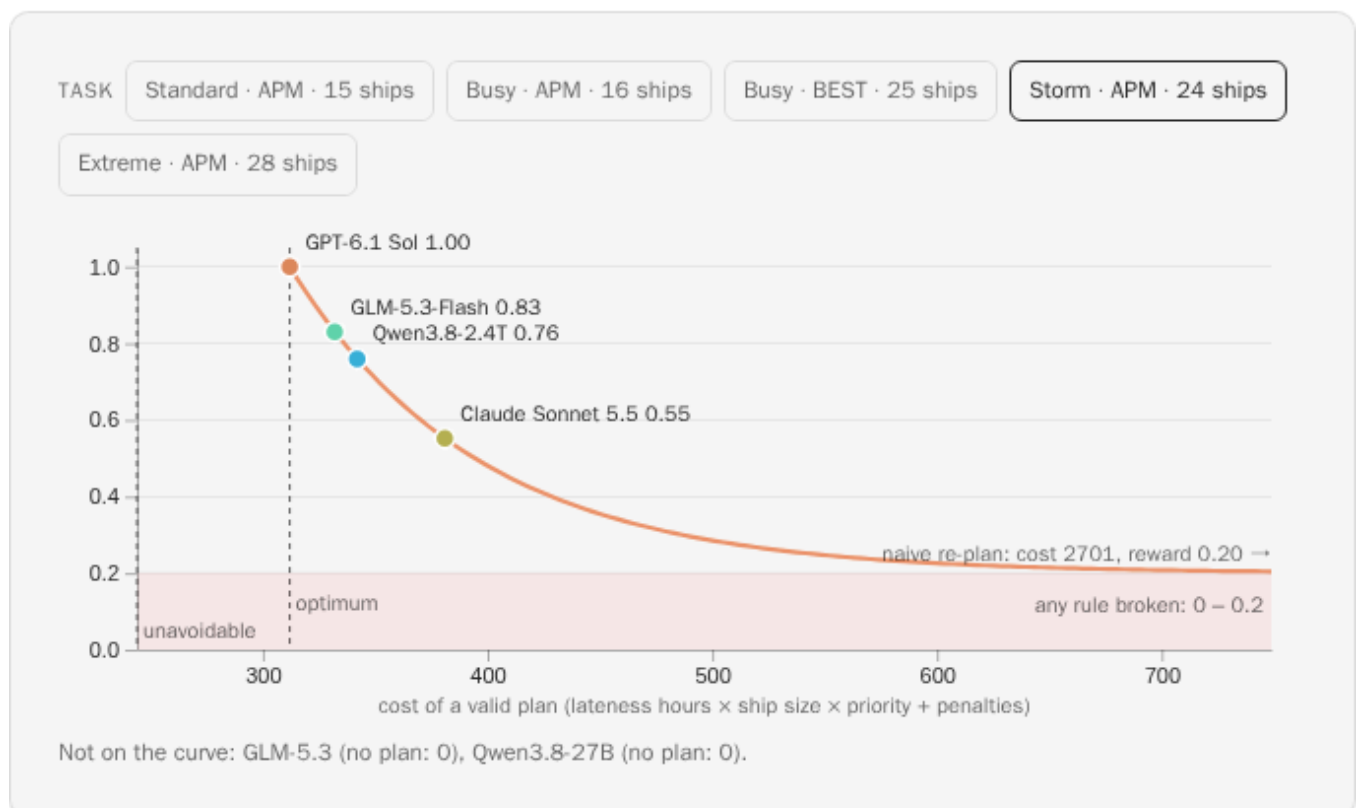
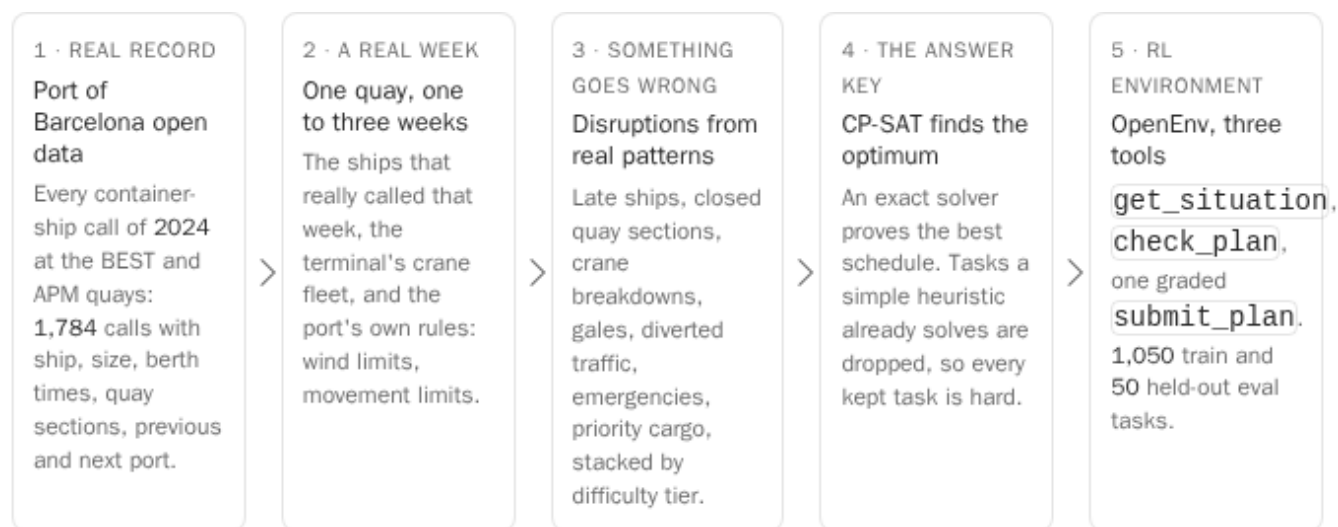


Figure 10 · 1.0 at the proven optimum, falling towards 0.2 as the plan gets costlier; plans that break a rule sit in the shaded band. Measuring the gap against the avoidable cost means a week full of unavoidable delay is not punished.

An independent checker re-verifies every plan. Across the 1,100 tasks the naive re-plan never scores above 0.44 and a greedy heuristic never above 0.82, because tasks they nearly solve were dropped when the dataset was built.

In the film's run, GLM-5.3 checks three drafts that come back with 9, 5 and then 0 rule breaks. Its last edits bring NAGOYA EXPRESS down from eight cranes to its six-crane limit and move CMA CGM MONTREAL from hour 131 to 133. The plan it submits costs 87: 72 in weighted delay plus three berth changes at 5 each. That equals the proven optimum, so the gap is zero and the reward is $0.2 + 0.8 \times e^0 = 1.00$.

Built on real Port of Barcelona data



Each task keeps a real week at one quay and the ships that really called, adds disruptions modelled on real events, and gets an answer key from CP-SAT. I drop the tasks a simple heuristic nearly solves, which leaves 1,050 training and 50 eval tasks. No week appears in both.

What is real and what is simulated. The records give us the berth plan the port actually ran, with the ships, their sizes, times and quay sections. They don't include crane assignments, container manifests or time spent waiting at anchor. Workloads, disruptions and the physical handling are simulated, and the film replays an agent's plan on that simulation.

From seed data to tasks

1. Seed: the real calls of one quay for one to three weeks, with the berth plan the port actually executed.
2. Fill the gaps: crane fleets, wind and traffic rules, and a crane rate of 28 moves per hour turn each call's real time alongside into a workload.

3. Disturb: inject the disruptions below, scaled by tier (standard, busy, storm, extreme).
4. Solve: CP-SAT proves the optimal re-plan, the answer key; the naive and greedy re-plans set the floor.
5. Filter: drop tasks a heuristic nearly solves and keep whole weeks out of training for the eval.

Disruption	What happens	Real-world model	Share of the 1,100 tasks
Late ships	ships arrive hours or days late	Red Sea diversions, <u>half of ships late</u>	100%
Closures	quay sections closed for repairs or dredging	crane-rail repair, fender repair, dredging	100%
Crane outages	cranes out of service for hours or days	breakdowns, maintenance	89%
Priority cargo	some ships count $\times 3$	transshipment connections	86%
Emergencies	a ship must dock by a deadline	a reefer ship losing power	76%
Extra calls	unscheduled ships need a berth	ad-hoc calls	63%
Gales	no ship moves; ≥ 300 m stop above 25 kn, all above 30	the port's <u>traffic ordinance</u>	62%
Bunching	several ships arrive together	delayed services catching up	46%
Diverted traffic	ships moved from the other terminal	terminal outages	15%

Data sources

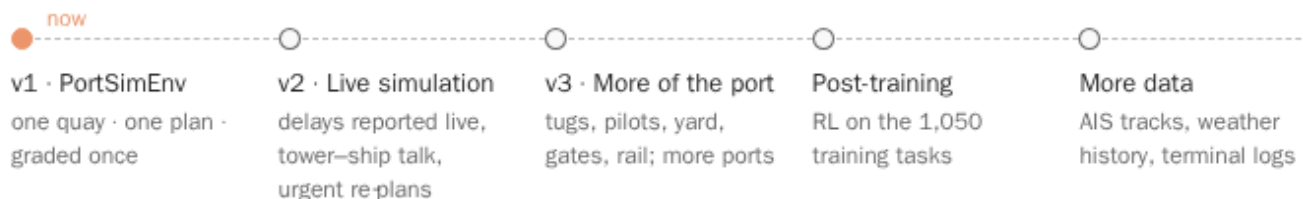
Data	Source	Licence
2024 ship calls: 1,784 at BEST and APM, with times, sections, ports	Port of Barcelona open data via berth-allocation-problems	CC BY-SA 4.0
Crane fleets: 13 at BEST, 9 at APM	BEST , APM Terminals	published
Wind, traffic and berth rules	BOE-A-2023-6719 , BOE-A-2025-5157 , port-call procedure	public
3D twin: coast, buildings, yards, cranes	OpenStreetMap	ODbL 1.0
Terrain	Terrain Tiles on AWS	open
Anchorage and routes (checked)	Copernicus Sentinel-2	open

Links

Environment: the OpenEnv Space, with the editor and 3D view and an MCP playground	FineEnvs/PortSimEnv
Eval: all 300 rollouts, replayed in 3D	FineEnvs/PortSimEnv-Eval
Dataset: tasks (train, eval), the source calls, the eval rollouts	datasets/FineEnvs/PortSimEnv
Bucket: 3D twin data and raw eval rollouts	buckets/FineEnvs/PortSimEnv
Collection	Simulation RL Envs
Code	github.com/adithya-s-k/FineEnvs: 07-simulation-environments/portsim-v1
Discussion	GitHub discussion #36

Contains data from the Port de Barcelona open data portal. The extracted data and the task pack built from it keep the source's CC BY-SA 4.0 licence.

Ongoing work



The question behind the series is still open. We can build realistic environments from seed data, but how do we get them as close to the real world as possible? If you have ideas for v2 or v3, for post-training, for other data sources, or a job worth simulating, I'd like to hear them in the [GitHub discussion](#).

Citation

For attribution in academic contexts, please cite this work as

Adithya S Kolavi (2026). "Simulation RL Environments".

BibTeX citation

```
@misc{kolavi2026_simulation_rl_environments,  
  title={Simulation RL Environments},  
  author={Adithya S Kolavi},  
  year={2026},  
}
```

Made with  with [research article template](#)