CAMP–II



Coding Agent Memory Post-Training II

Unlocking the Memory Potential of Pre-trained File Operations for Long-Horizon Tasks via On-Policy Distillation

Lirui Luo · Kelong Mao · Xinwei Yang · Rongqing Li · Luyu Chen · Kieran Wong
Yudong Guo · Xinrui Wang · Jiayin Zhu · Pu Yao · Gu Simiu · Sulong Xu · Cong Fang

Peking University · JD.com · Sichuan University
Beijing Institute of Technology · Renmin University of China · The Hong Kong University of Science and Technology

Teaching agents to remember through files.

liruiluo.github.io/agentmemorygym-v2/

Paper

The context ends. The task does not.

Task: build a fitness-product showcase with products, collections, picks and comparisons.

Before the boundary

Styles are present.
The HTML body is missing.

CONTR.txt

CSS only. No HTML body.
Rewrite all four sections.

After the boundary

The history is gone.
Read the plan and continue.

Context is replaced

The file persists

The memory must preserve what the next context needs.

Visual case · explanatory schematic; actual trace on slide 12

A strong teacher still needs a usable memory interface

On-policy distillation learns from actions the student actually samples.

Student side

Sample a meaningful memory action.

If a write never appears,
it receives no supervision.

Teacher side

Give useful feedback on that same action.

A strong task model may still be
unfamiliar with the memory protocol.

Memory learning depends on both action coverage and teacher feedback.

Use the file operations both models already know

A shared Bash interface, an episode-persistent workspace, one task-acting policy.

Create / revise → Search / read → Continue

Save requirements, findings and a plan.

Recover relevant state when it is needed.

Use that state for the next task decision.

Episode-persistent workspace

`notes.md` `evidence/` `CONTINUATION.md`

Familiar operations provide the channel. Training teaches the memory.

The student acts. The frozen teacher scores.

The teacher sees the student’s recorded context and scores the response it sampled.

01

Sample

Task actions and file writes come from the student.

→

02

Score

Scores the same tokens in the same context.

→

03

Update

Train the student on both task and memory tokens.

Token feedback = teacher log-probability − student log-probability

Learn from the teacher during training. Act without it at inference.

One policy learns across five task worlds

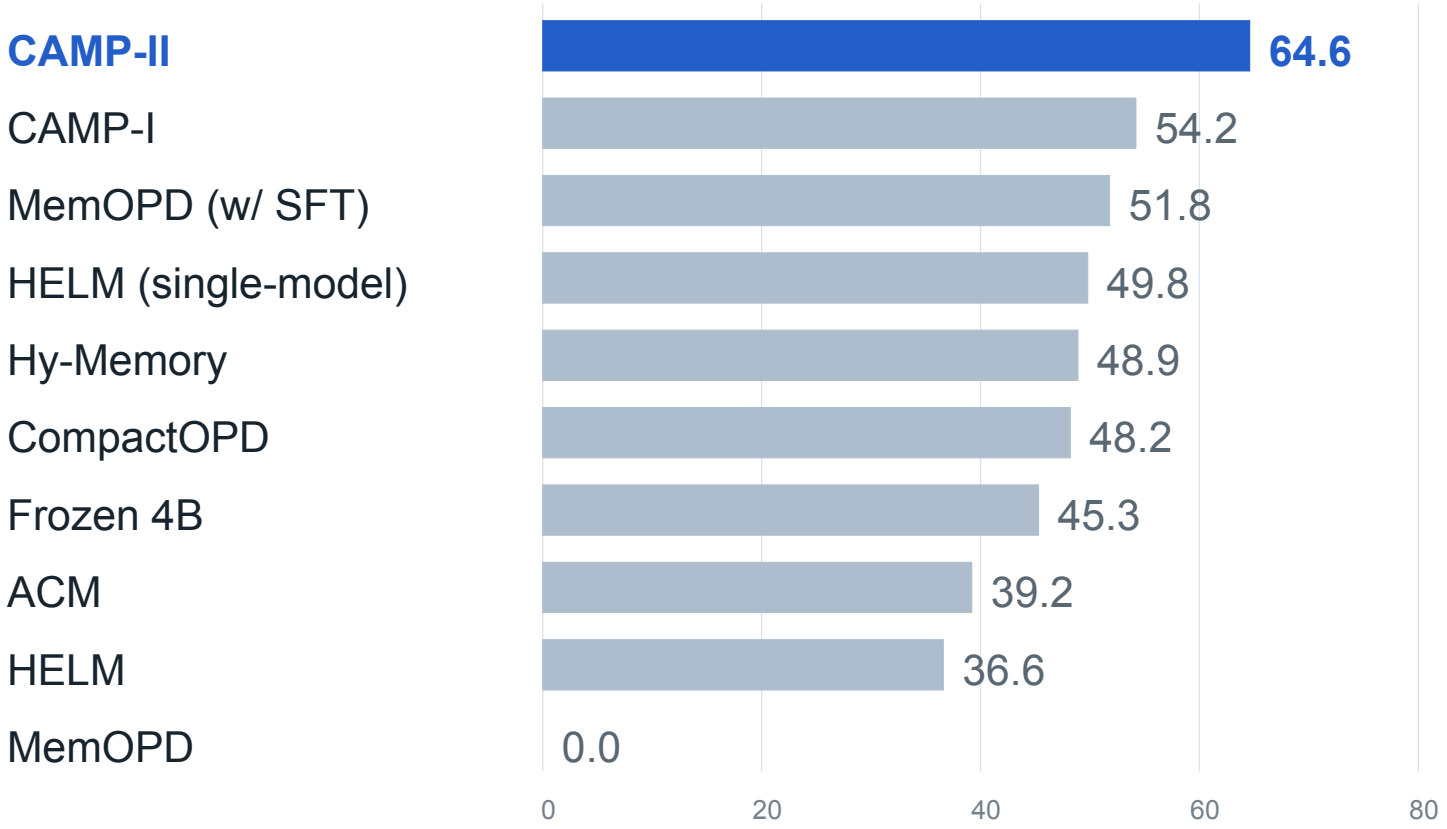
Native task semantics stay intact. The file-memory interface is shared.

Shop	Coding	AutoResearch	General	Visual
Six linked purchases	Repository repair	ML experiment iteration	Multi-tool knowledge work	Website development
Product evidence	Diagnosis + tests	Results + next trial	Requirements + outputs	Visual findings + plan

One joint policy · ordinary Bash · persistent workspace

File-based OPD gives the largest average gain

The full 4B comparison set. CAMP-I uses task RL; the memory baselines use OPD adaptations.



+12.8

points over the strongest memory OPD baseline

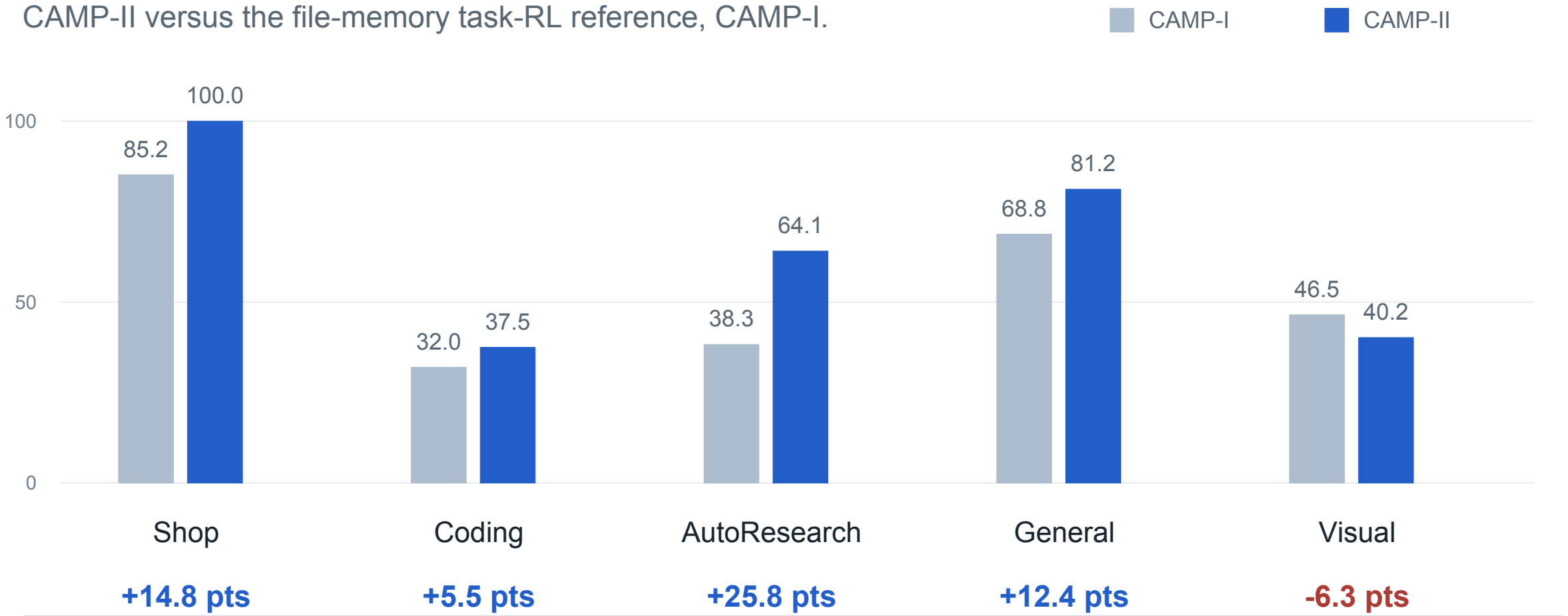
2.97× the gain

From the same initial 4B: +19.3 vs +6.5 points.

Table 1 · five-domain mean: four success rates + Visual panel-mean judge score

The average improves; Visual remains a trade-off

CAMP-II versus the file-memory task-RL reference, CAMP-I.



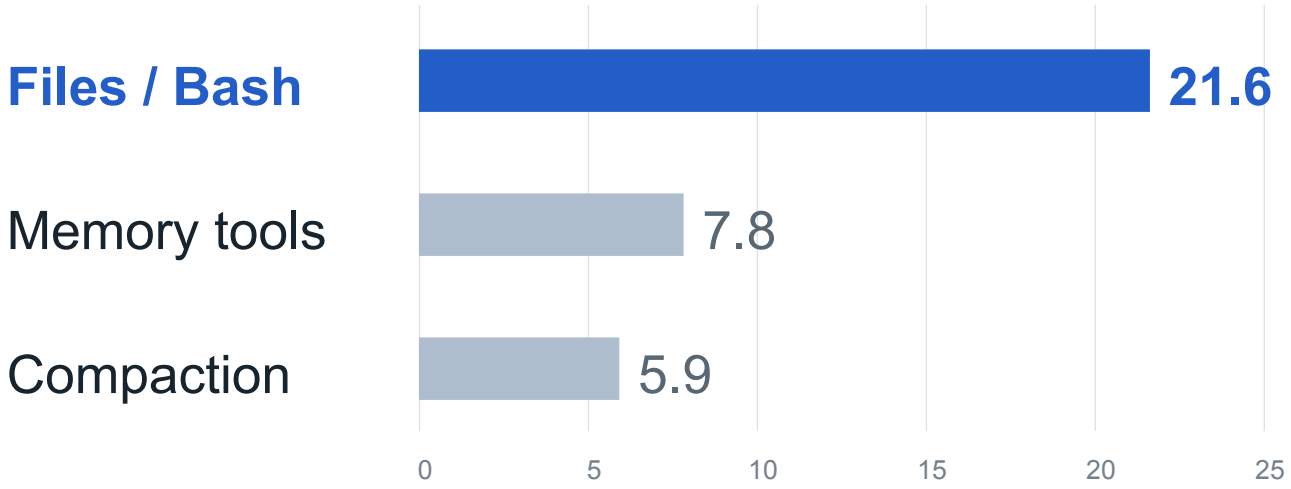
The largest gain over CAMP-I is in AutoResearch: +25.8 points.

Table 1 · Visual is a panel-mean judge score; other domains use task success

Files expose more memory actions to learning

The file interface is familiar to both models and used more by the student.

Memory actions / all actions (%)



Lower surprisal

Nats per token · lower is more familiar

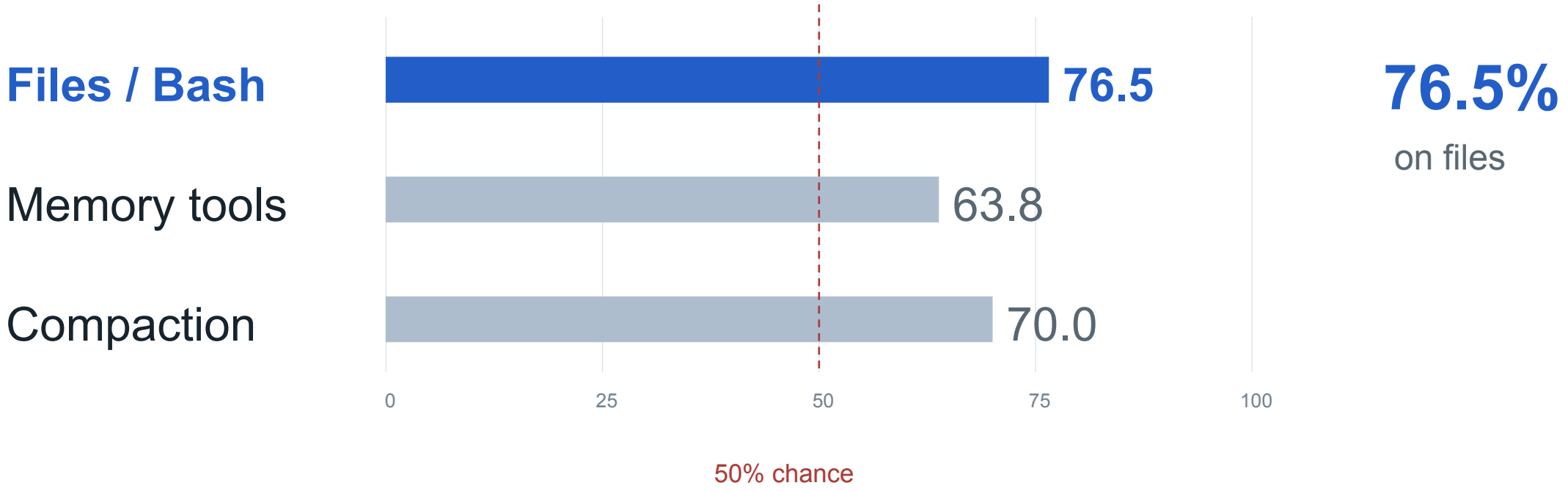
	Teacher	Student
Files	**0.42**	**0.19**
Tools	0.91	0.70
Compact	1.40	0.98

A familiar channel gives OPD more memory actions to supervise.

Figure 3 · action share and frozen-model surprisal

The teacher better recognizes task-matched notes

Does the teacher prefer a note from this task over a note from another task?



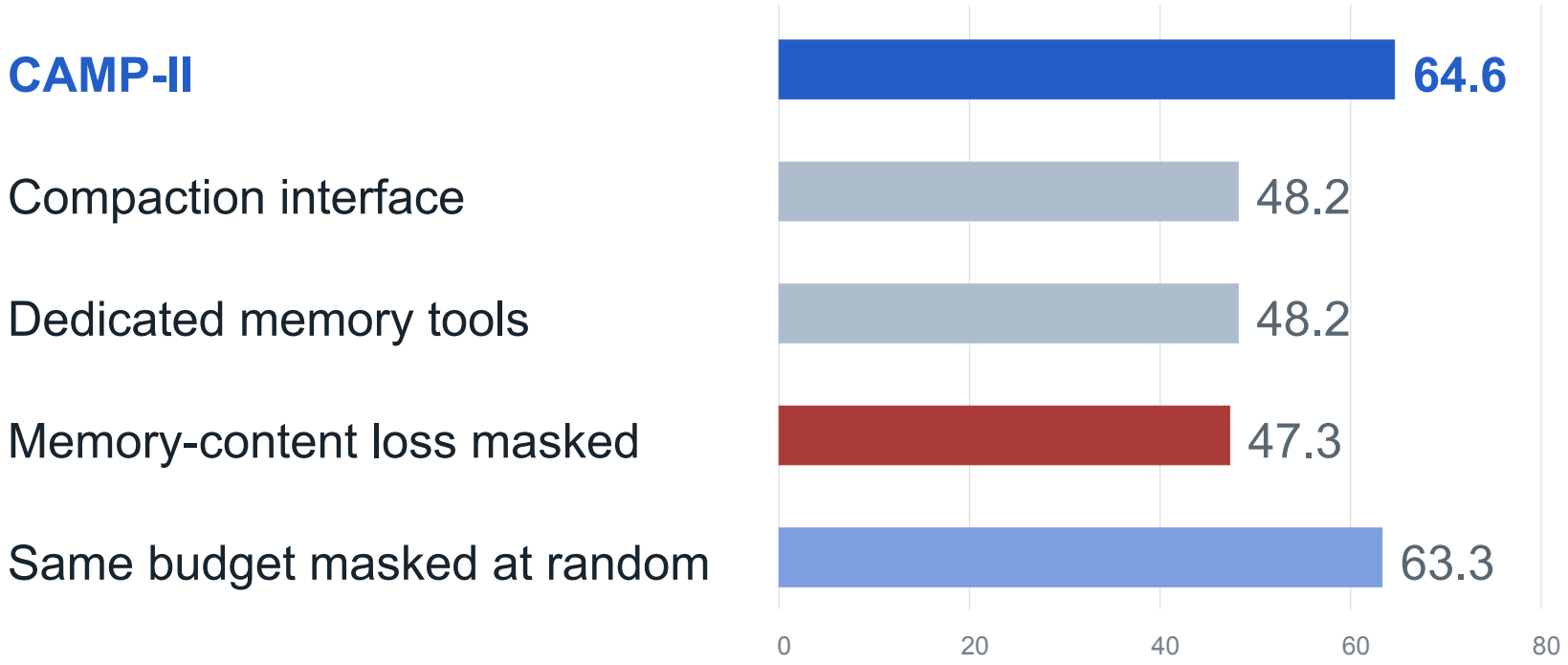
76.5%

on files

This tests task relevance. Next: isolate what the student learns.

Learning what to write carries the gain

Mask memory-content supervision, or mask the same token budget at random.



−17.3

memory-content loss masked

−1.3

same budget masked at random

Which tokens are taught matters—not only how many.

Figure 6 · controlled interface and loss-masking ablations

The agent resumes from a plan it wrote

Task: Build a fitness-product showcase

A static gallery with product cards, collections, comparisons and customer reviews.

(a) Write task memory

`CONTR.txt` — Turn 32

Problem

“a malformed CSS-only file”

“NO HTML body markup”

Plan

“Complete rewrite of dist/index.html with all sections”

“Featured products” · “Trending collections” “Editor picks” · “Comparison cards”

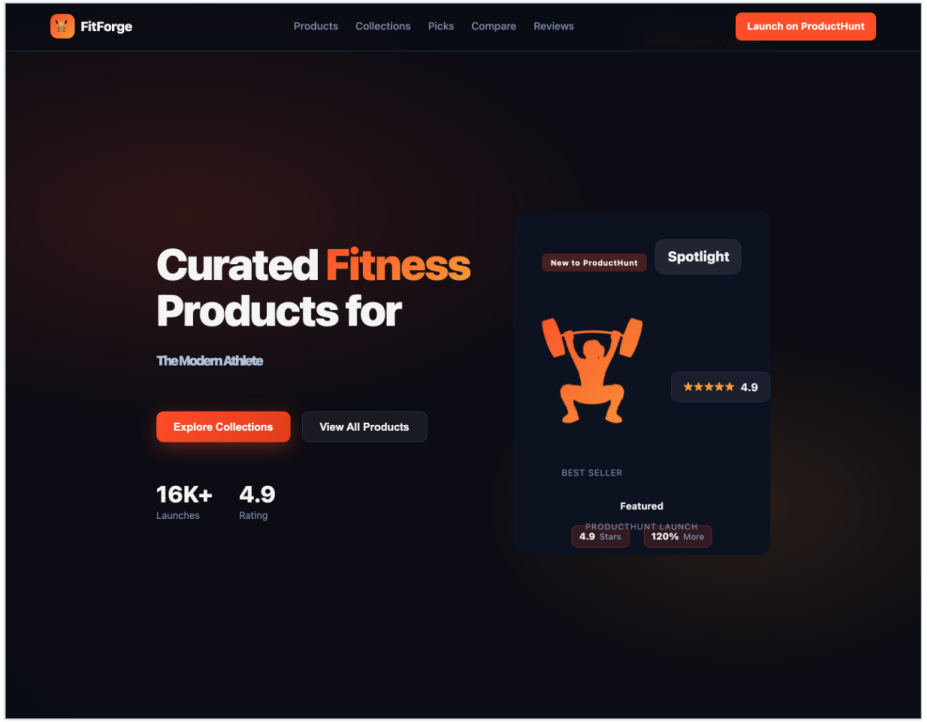
Context replacement · turn 33

(b) Retrieve task memory

Turn 35	`ls -la`
Turn 36	`cat CONTR.txt`

(c) Continue development

Turns 37–54



Write → replace context → read → continue

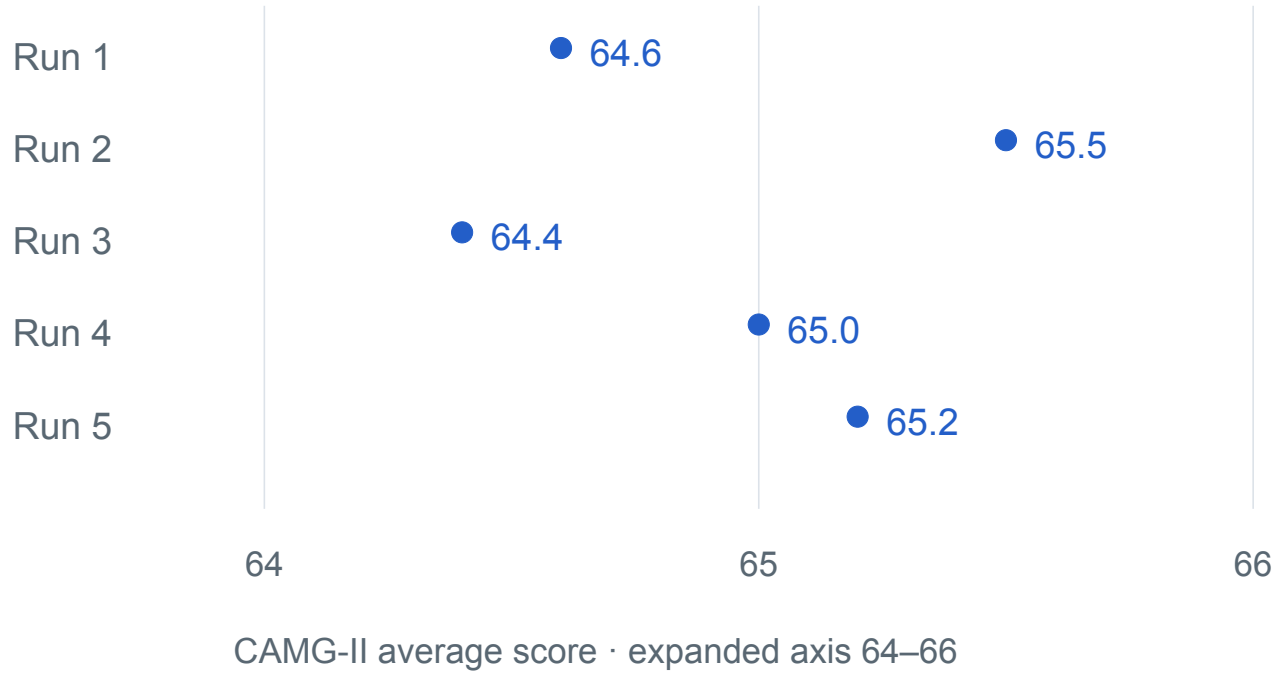
Paper Figure 5 · observed Visual training episode; not a causal intervention

The result repeats across five training seeds

Same CAMP-II recipe. Five independent training runs.

64.9 ± 0.5

Five-domain mean ± sample SD



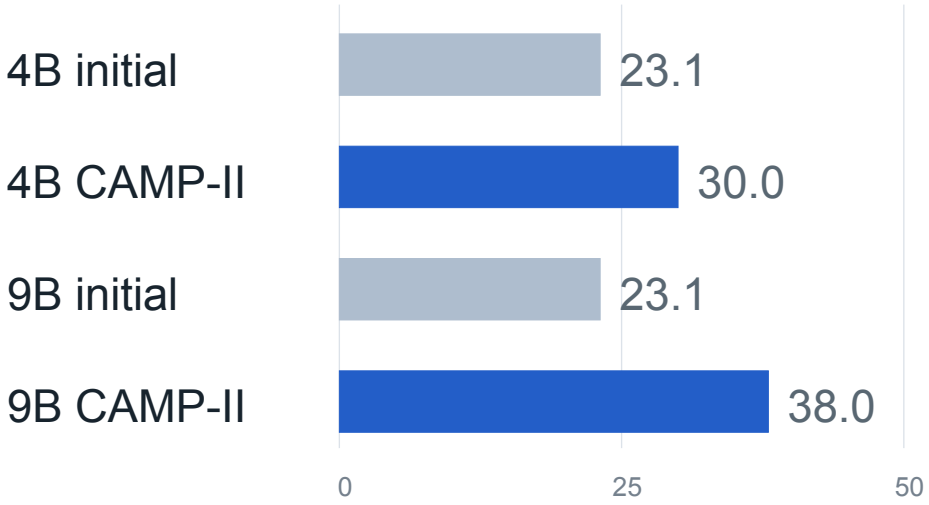
The five reported averages range from 64.4 to 65.5.

Appendix C.3 · sample standard deviation across training seeds

The learned policy improves on two external benchmarks

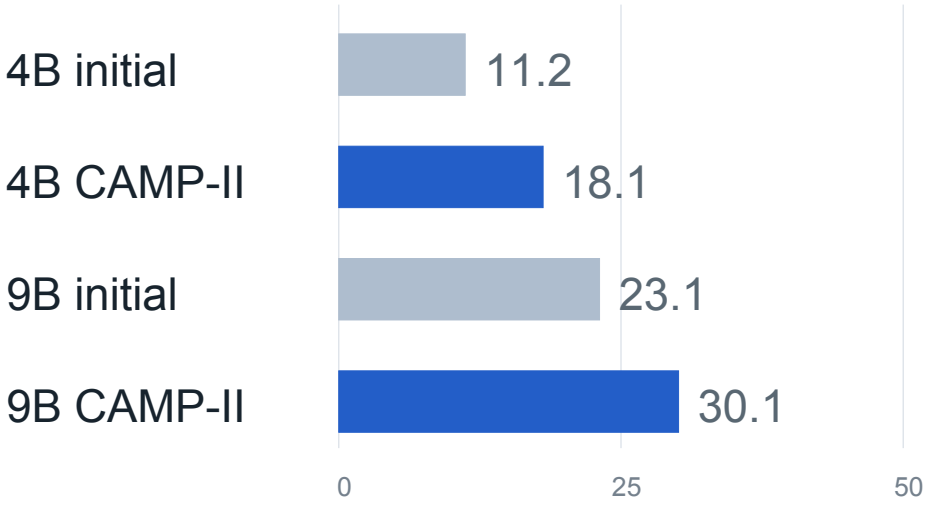
Both model sizes improve over the checkpoints they start from.

Terminal-Bench 2.1



35B-A3B reference: 33.8

SWE-Bench Pro V2



35B-A3B reference: 27.4

Memory learned in the gym carries into new task benchmarks.

Table 2 · success rates (%); larger reference is Qwen3.5-35B-A3B

Familiar operations make better teaching channels

Teach memory through files.

The principle	Match the memory interface to both the student and the teacher.
The method	Distill the task policy’s own file actions with token-level feedback.
The evidence	Higher average scores, memory-specific ablations and external transfer.

One shared file-memory interface. Five long-horizon task worlds.