Training One Model to Master Cross-Level Agentic Actions via Reinforcement Learning

Kaichen He[1], Zihao Wang[1], Muyao Li[1], Anji Liu[2], Yitao Liang[1]
[1]Peking University, [2]National University of Singapore

Motivation:

Agentic AI is evolving:
- From engineered workflows to post-training native models.

Existing agents are limited by fixed action spaces:
- GUI, API, MCP scripts or robotic commands are usually predefined.
- Limits flexibility and generalization in dynamic environments.

Optimal action granularity varies across tasks and within tasks:
- Some steps need high-level API actions for efficiency.
- Other steps need low-level atomic actions for precision.

Key challenge:
- How to build an agent that dynamically chooses the best action space for each step?

Motivation

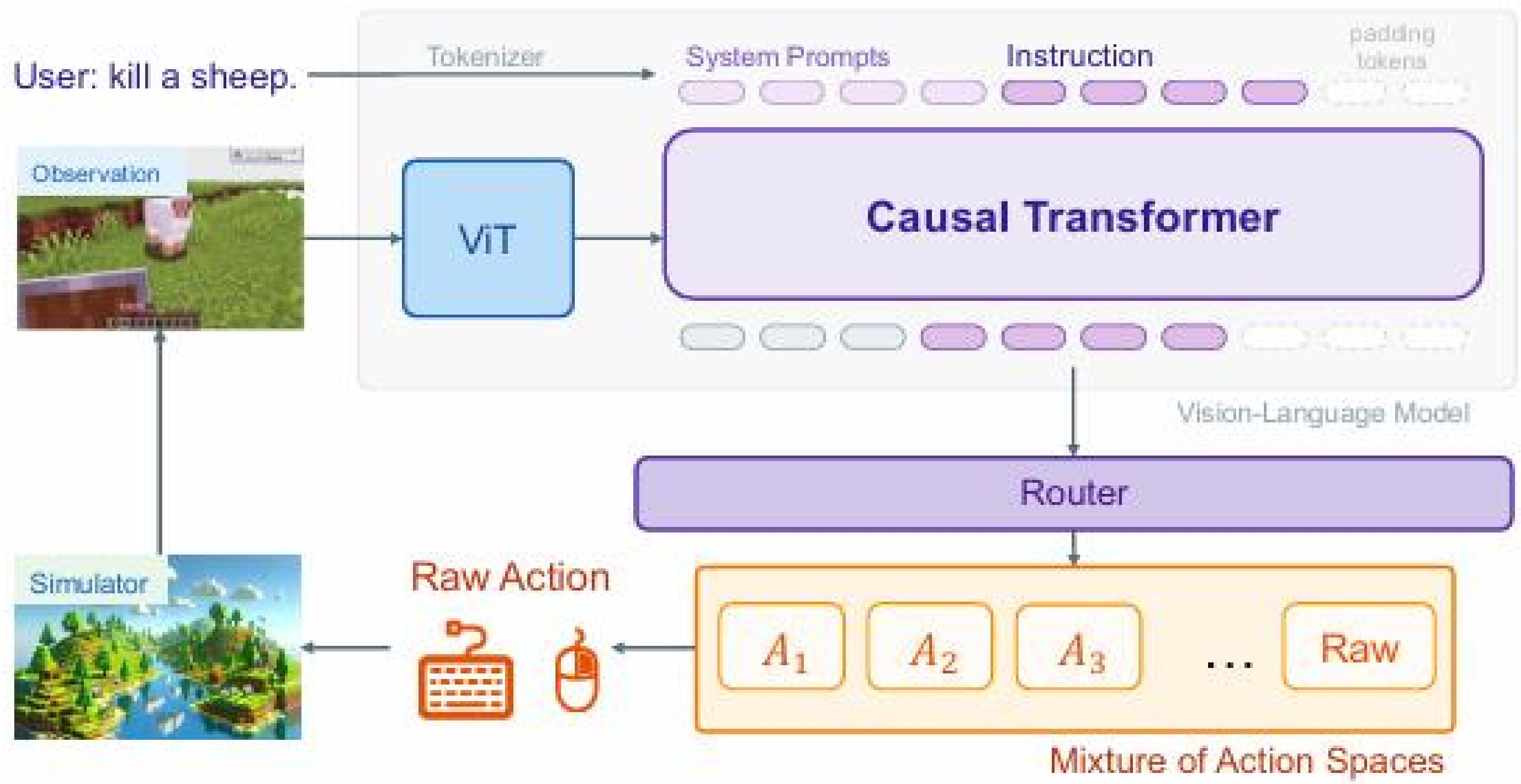


Figure 1 | **The CrossAgent Framework**. Unlike prior methods that confine the agent to a fixed action space (e.g., atomic movements) throughout a trajectory, CrossAgent dynamically switches across different action spaces to adapt to the context.

Method: A Three-Stage Pipeline

- **Stage I: <u>Mixed-Space Supervised Finetuning</u>**

Learn to execute heterogeneous actions.

Train on balanced trajectories from multiple action spaces to build a unified cross-level policy.

- **Stage II: <u>Single-Turn Reinforcement Learning</u>**

Learn which action space fits the local context.

Optimize one-step action selection with action-space-agnostic rewards.

- **Stage III: <u>Multi-Turn Reinforcement Learning</u>**

Learn long-horizon switching strategies.

Optimize action-space switching across trajectories with success rewards and action-cost penalties.

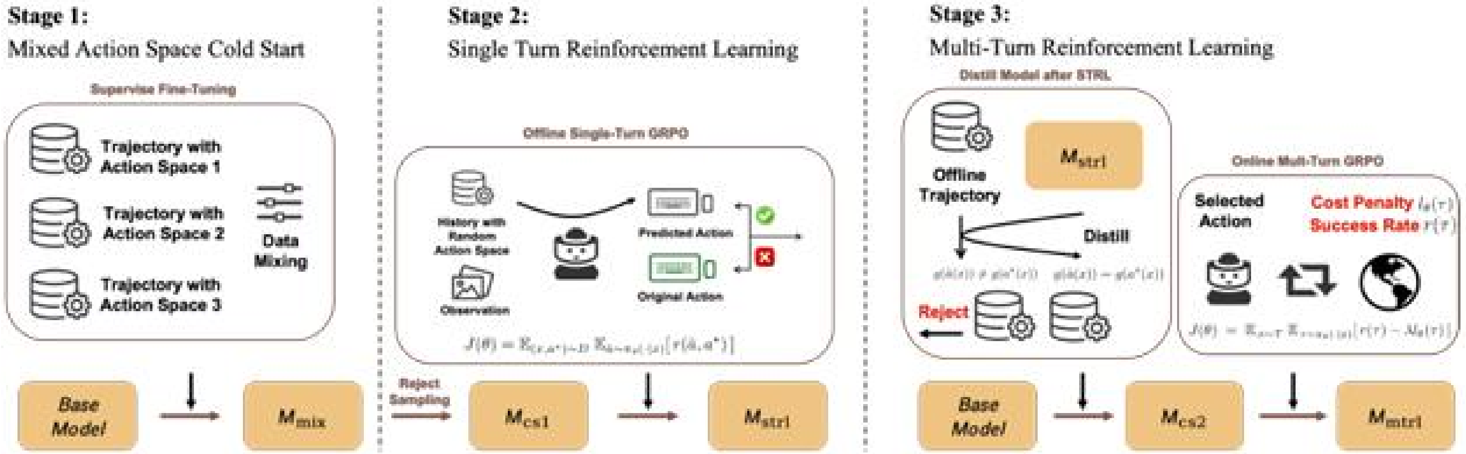


Figure 2 | **Overview of the CrossAgent Training Pipeline.** The pipeline comprises three distinct stages: Cold-Start Supervised Fine-Tuning (SFT), Single-Turn Reinforcement Learning (STRL), and Multi-Turn Reinforcement Learning (MTRL). In the first stage, the model learns to decode actions from a heterogeneous action space using a balanced dataset. During STRL, the model is fine-tuned to autonomously select the appropriate action space based on the immediate task context. Finally, in the MTRL stage, the policy is further optimized to balance task success rate with execution efficiency over long horizons. This progressive pipeline ensures CrossAgent effectively adapts its action granularity across a wide range of tasks.

Different tasks and phases favor different action spaces.

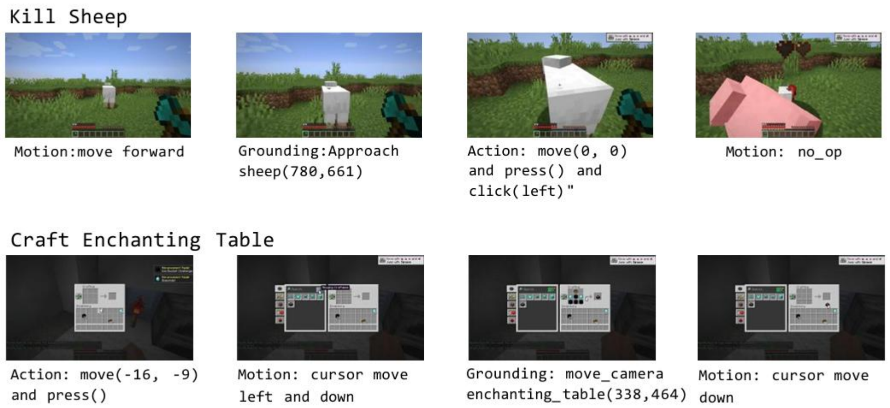
Naive mixed-space training exposes agents to diverse actions, but does not teach reliable action-space selection.

Capability → Local Preference → Long-Horizon

The Training Strategy enabling agents to learn not only how to act, but when to use each action space.

Experiment

CrossHA RL finetuned on 30 tasks achieves state-of-the-art performance across 800+ Minecraft tasks involving both embodied control and GUI interaction.

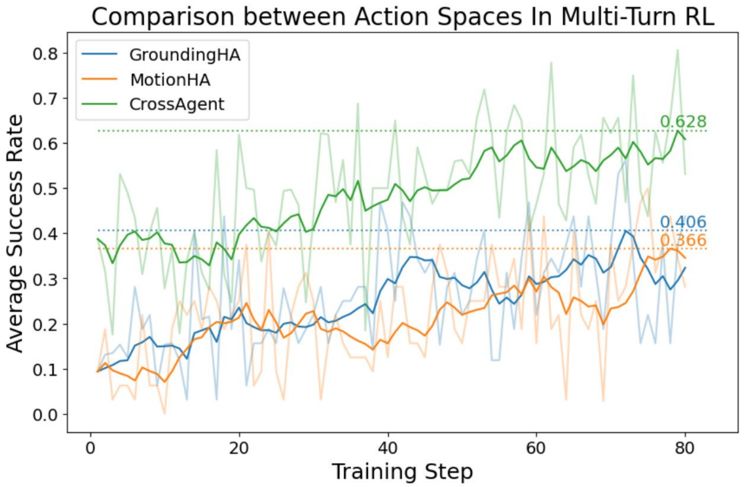


Method	Mine Blocks			Kill Entities			Craft Items			All Tasks	
		FT ↑	*ASR* ↑		*FT* ↑	*ASR* ↑		*FT* ↑	*ASR* ↑	*FT* ↑	*ASR* ↑
Instruction-Conditioned Policies											
VPT [4]	20.0	30.7	$6.0^{\pm 11.4}$	10.0	24.6	$3.6^{\pm 7.7}$	0.0	6.7	$0.8^{\pm 3.3}$	20.7	$3.5^{\pm 8.4}$
STEVE-I [26]	50.0	29.4	$8.0^{\pm 17.0}$	0.0	14.7	$3.9^{\pm 12.0}$	0.0	16.4	$3.2^{\pm 8.4}$	20.2	$5.0^{\pm 12.4}$
ROCKET-1 [11]	60.0	57.5	$18.9^{\pm 24.3}$	60.0	**63.9**	$27.9^{\pm 29.3}$	0.0	0.0	$0.0^{\pm 0.0}$	45.5	$15.6^{\pm 24.9}$
JARVIS-VLA [25]	55.0	55.3	$30.0^{\pm 35.4}$	60.0	61.9	$18.5^{\pm 22.7}$	40.0	**74.3**	$25.1^{\pm 23.9}$	**63.8**	$24.5^{\pm 28.4}$
VLM-based Agents											
TextHA [49]	60.0	36.3	$27.2^{\pm 38.2}$	0.0	19.2	$8.7^{\pm 23.7}$	50.0	43.9	$26.0^{\pm 35.2}$	33.1	$20.6^{\pm 34.0}$
GroundingHA [11]	**90.0**	**61.0**	$37.1^{\pm 38.5}$	50.0	**90.1**	$26.5^{\pm 23.4}$	15.0	27.5	$6.7^{\pm 10.8}$	59.5	$23.4^{\pm 29.6}$
UI-TARS-1.5 [36]	-	-	$42.1^{\pm 20.4}$	-	-	$31.0^{\pm 16.4}$	0	0	$36.7^{\pm 17.2}$	-	33.8
MotionHA [6]	70.0	51.0	$27.4^{\pm 35.2}$	20.0	29.5	$4.3^{\pm 10.8}$	0.0	0.0	$0.0^{\pm 0.0}$	29.8	$10.6^{\pm 24.4}$
LanguageHA [14]	60.0	31.3	$11.3^{\pm 14.5}$	0.0	12.8	$6.5^{\pm 9.3}$	5.0	19.3	$6.3^{\pm 9.2}$	21.1	$8.0^{\pm 11.5}$
LatentHA [47]	70.0	54.2	$24.4^{\pm 31.1}$	50.0	24.6	$8.5^{\pm 17.9}$	0.0	19.1	$3.0^{\pm 7.5}$	32.6	$12.0^{\pm 23.0}$
OpenHA [49]	80.0	**67.3**	$30.1^{\pm 13.9}$	**70.0**	62.6	$32.5^{\pm 9.2}$	**80.0**	58.8	$31.9^{\pm 13.7}$	**62.8**	$31.5^{\pm 12.5}$
Game-TARS [50]	-	-	$50.14^{\pm 20.7}$	-	-	$38.1^{\pm 24.6}$	-	-	$39.1^{\pm 27.5}$	-	42.2
Ours											
CrossHA(w/o STRL stage)	63.6	35.1	$39.0^{\pm 46.5}$	28.6	43.9	$27.7^{\pm 43.9}$	57.1	55.7	$58.0^{\pm 48.4}$	44.9	$41.6^{\pm 47.9}$
CrossHA	**94.7**	45.2	$40.0^{\pm 48.3}$	**66.6**	58.1	$45.1^{\pm 43.5}$	**83.3**	**72.7**	$78.8^{\pm 41.0}$	58.7	$54.6^{\pm 47.6}$

Experiment

- Mixed action spaces matter;
- STRL matters;
- High OOD generation ability;

Both heterogeneous action spaces and step-level action-space selection are crucial for efficient long-horizon RL.



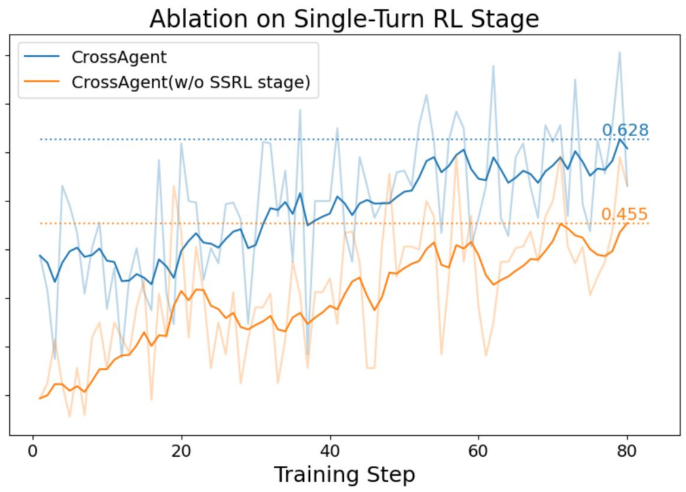


Table 2. Evaluation results of RL agents on In-Distribution (ID) and Out-of-Distribution (OOD) tasks. We report the success rate and standard deviation. **Red** indicates the best performance, and **Blue** indicates the second best.

Method	In-Distribution Evaluation				Out-of-Distribution Evaluation			
	Mine Blocks	Kill Entities	Craft Items	All Tasks	Mine Blocks	Kill Entities	Craft Items	All Tasks
RawHA-RL	$65.2^{\pm32.1}$	$41.6^{\pm21.8}$	$96.2^{\pm3.8}$	$70.1^{\pm33.6}$	$28.9^{\pm44.2}$	$28.4^{\pm32.3}$	$69.8^{\pm44.4}$	$42.4^{\pm45.1}$
GroundingHA-RL	$46.8^{\pm35.6}$	$48.8^{\pm34.5}$	$69.3^{\pm28.4}$	$52.6^{\pm31.5}$	$27.2^{\pm42.4}$	$33.9^{\pm35.9}$	$57.2^{\pm48.1}$	$39.4^{\pm44.3}$
MotionHA-RL	$78.6^{\pm24.6}$	$44.0^{\pm25.2}$	$63.0^{\pm20.7}$	$61.9^{\pm27.5}$	$41.5^{\pm45.5}$	$26.8^{\pm26.7}$	$49.0^{\pm47.3}$	$39.1^{\pm42.0}$
CrossHA(w/o STRL)	$63.3^{\pm28.9}$	$24.0^{\pm26.4}$	$76.1^{\pm27.2}$	$54.5^{\pm35.3}$	$39.0^{\pm46.5}$	$22.2^{\pm42.6}$	$58.0^{\pm48.4}$	$39.7^{\pm48.1}$
CrossHA	$70.7^{\pm33.3}$	$52.1^{\pm22.9}$	$83.7^{\pm25.5}$	$68.8^{\pm30.5}$	$40.0^{\pm48.3}$	$28.4^{\pm33.2}$	$78.8^{\pm41.0}$	$49.1^{\pm46.6}$

Case Studies:

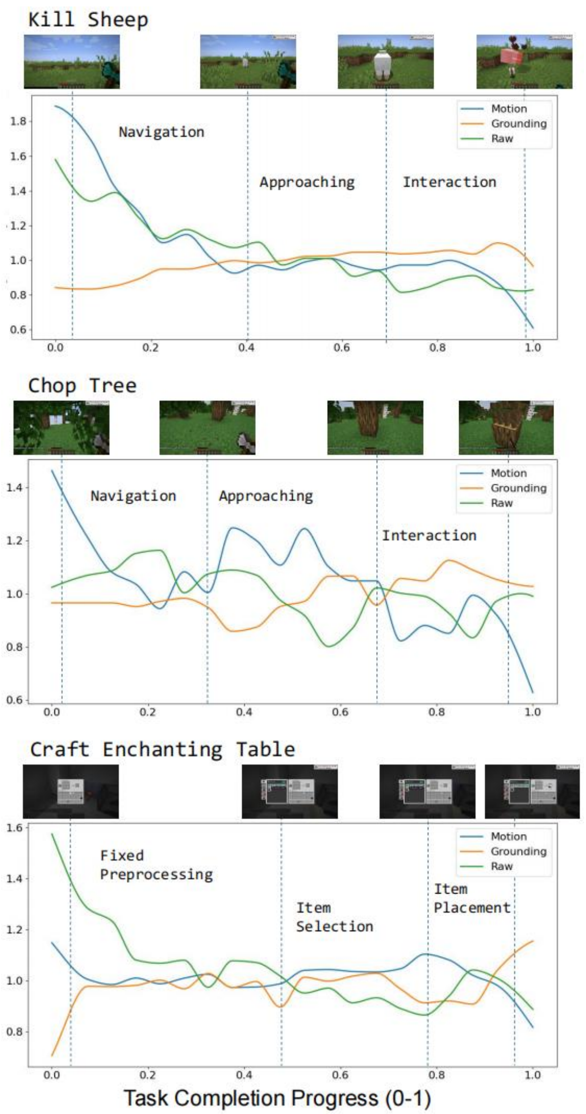
•Navigation / approaching: prefers motion-level actions for efficient movement.

•Interaction / item placement: increases grounding or raw actions for precise control.

•Task-dependent switching: different tasks show different action-space density patterns.

CrossHA learns when to use which action space, rather than relying on fixed action rules.

Probability Density of Action Spaces



Thank you !