Continual Learning for Vision

Manling Li
Northwestern University
Amazon Scholar
manling.li@northwestern.edu

NORTHWESTERN
UNIVERSITY

MLL LAB
Machine Learning and Language

How Models Learn (and Fail to Learn) the Physical World

A robot arm is placing the banana on top of the coaster

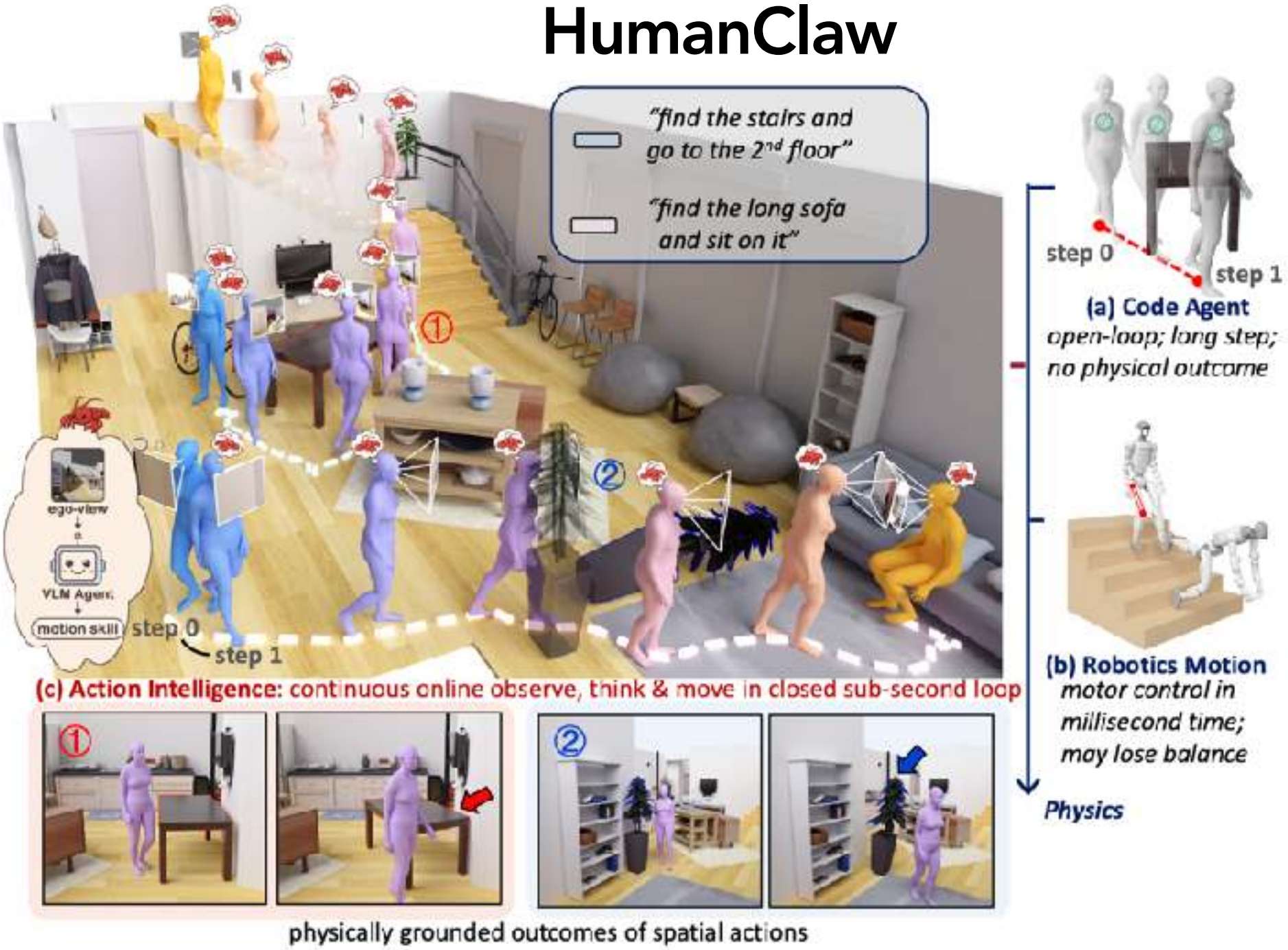
World Models $\overset{?}{==}$ **Surface Behavior**

what we want — what we truly learn

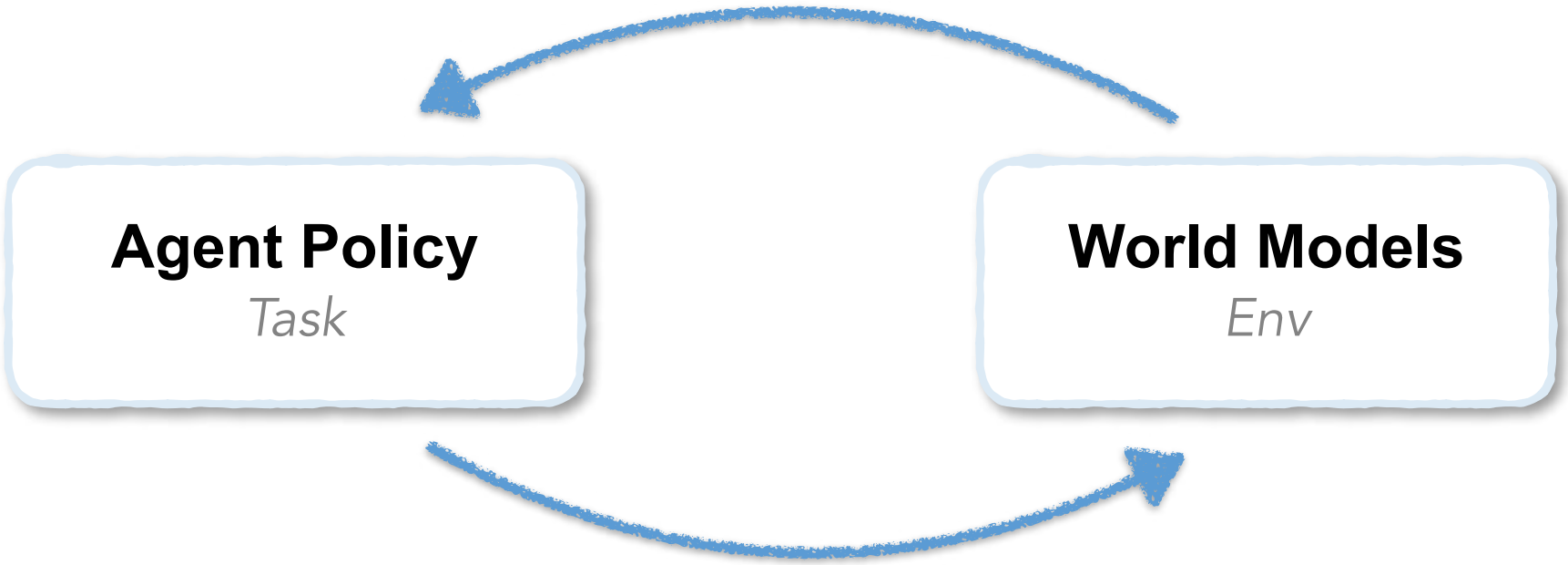
GPT-6

REAL - Exterior 1
REAL - Exterior 2
REAL - Wrist
SIM - Exterior 1
SIM - Exterior 2
SIM - Wrist
Hello燕Sir
小红书

HumanClaw


"find the stairs and go to the 2nd floor"
"find the long sofa and sit on it"
ego-view
VLN Agent
motion skill
step 0
step 1
step 0
step 1
(a) Code Agent
open-loop; long step;
no physical outcome
(b) Robotics Motion
motor control in
millisecond time;
may lose balance
Physics
(c) Action Intelligence: continuous online observe, think & move in closed sub-second loop
physically grounded outcomes of spatial actions

Co-Evolving: Let's train them!



World models are not about surface behavior,
but the reasoning that matters to the downstream agents

The Age of Agents

Capability

2026.01
OpenClaw becomes fastest growing repo in history

2025.11
Agentic browsers emerge

2024.11
Anthropic creates MCP

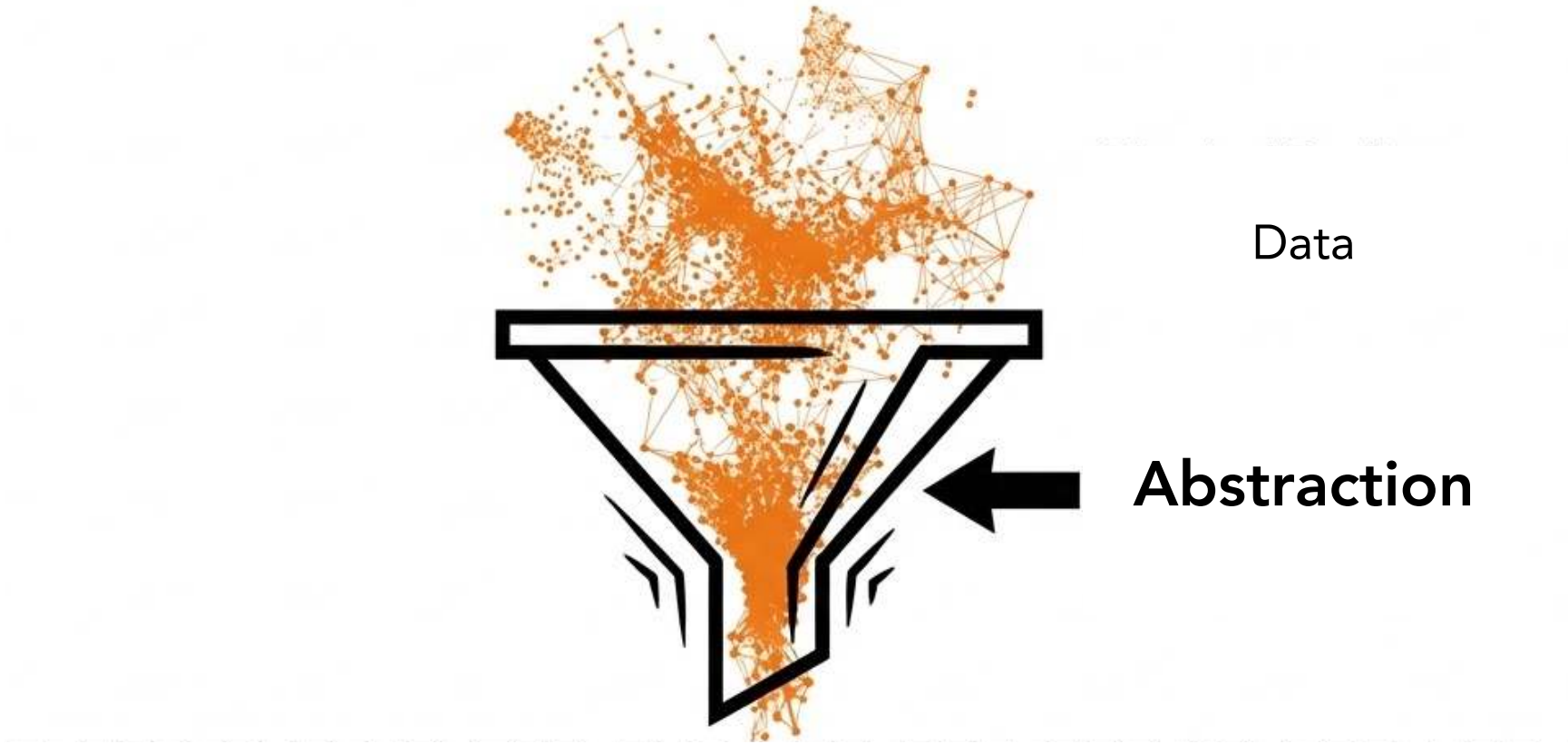
2023.03
AutoGPT & BabyAI

2023 **2024** **2025** **2026**

OPENCLAW

What's Left for Humans?

What is the missing intelligence: Abstraction

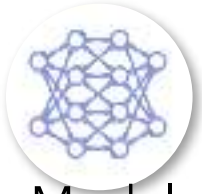


RSI: Recursive Self-Improvement?

Abstraction as Compression

Model vs Human:

Data ↑ Abstraction ↓

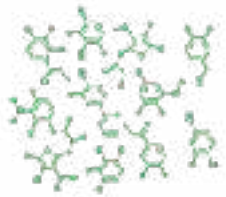


Model

1 Million Molecules

→ **Statistical Correlation**

Human



10 Molecules

→ **Chemical Law (Abstraction)**

On the measure of intelligence

The intelligence of a system is a measure of its skill-acquisition efficiency over a scope of tasks, with respect to priors, experience, and generalization difficulty.

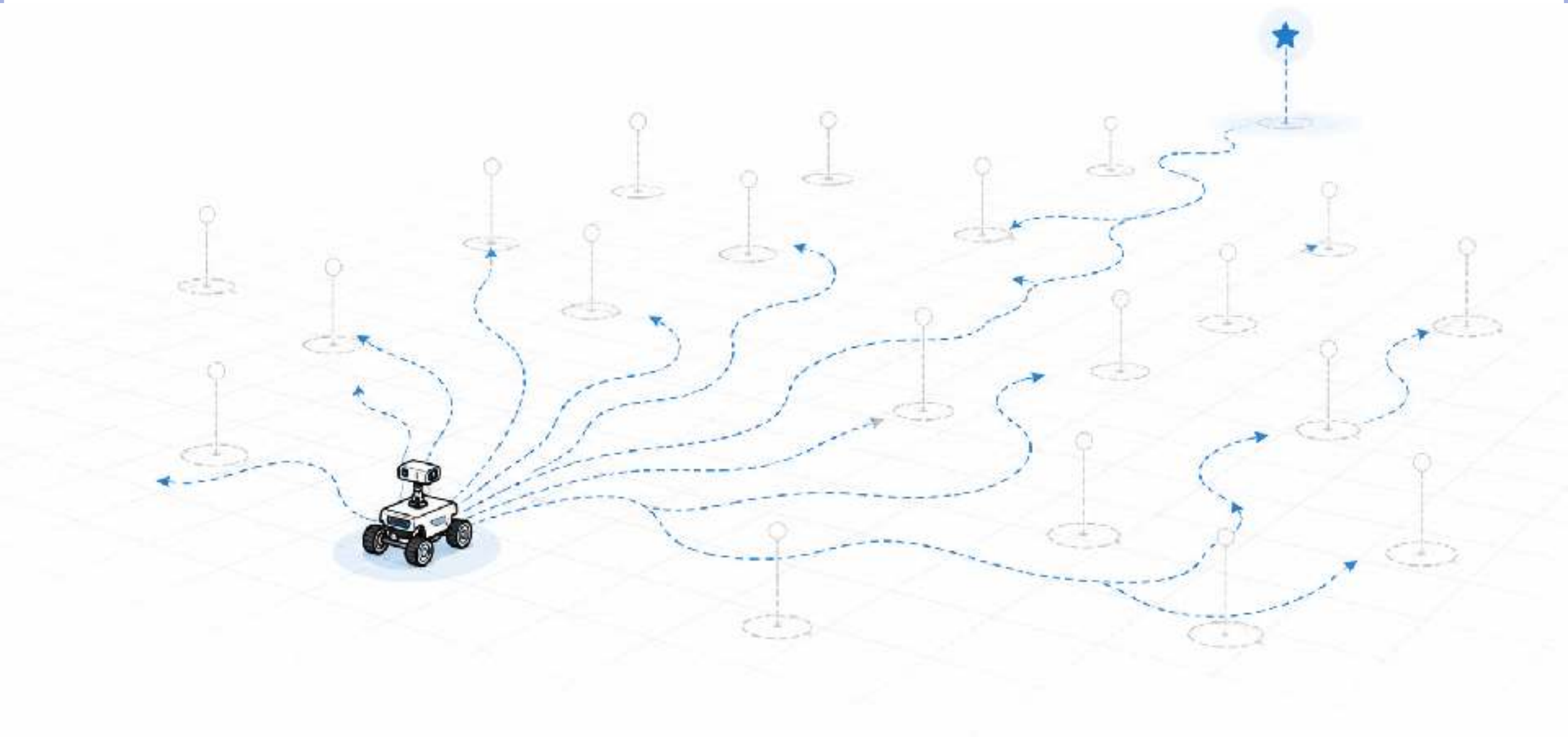
— Collet, et al. On the measure of intelligence, 2019.

Why is RAG not enough?

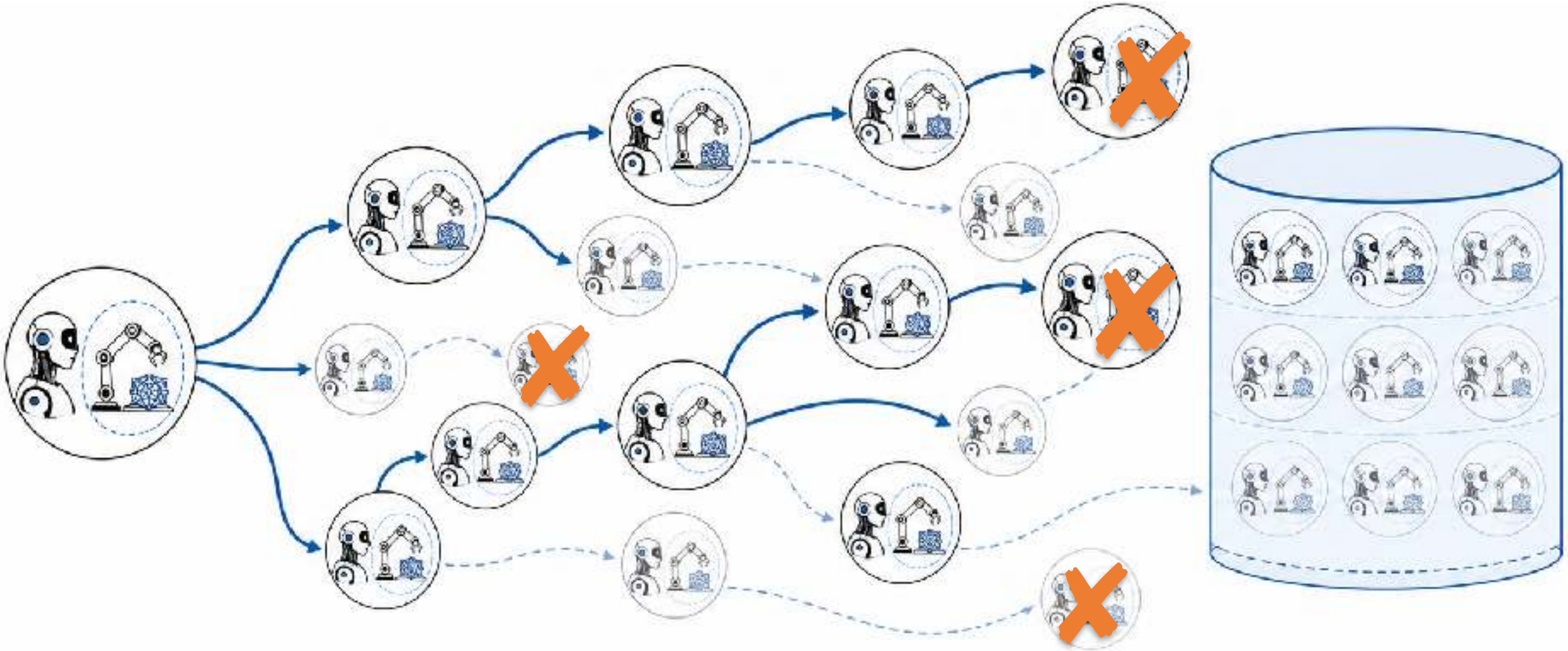
DeepResearch

AutoResearch

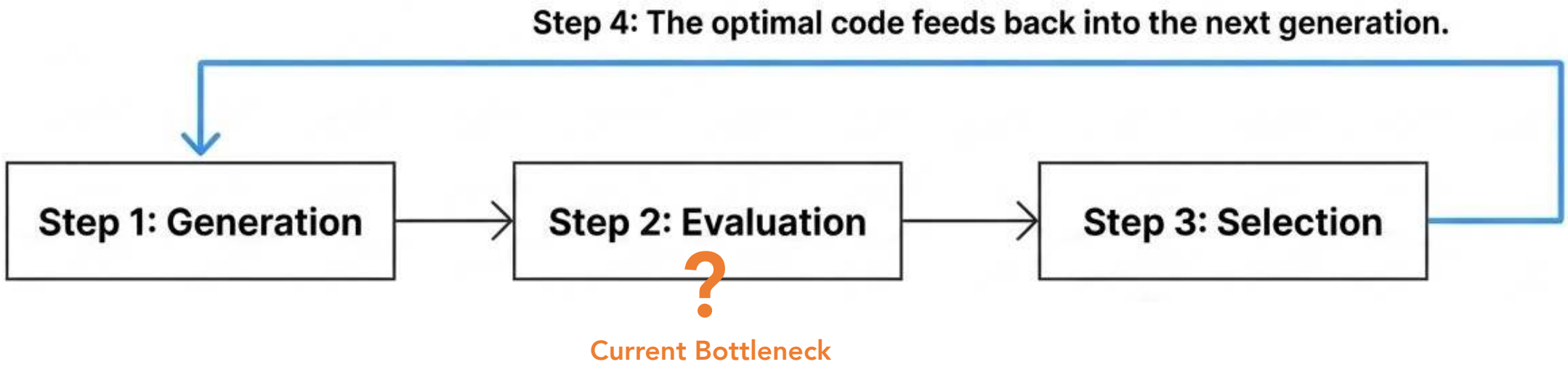
Why is RAG not enough?



Now: Recursive Self Improvement (RSI)



Recursive Self Improvement: AutoResearch



The Path to Recursive Self-Improvement

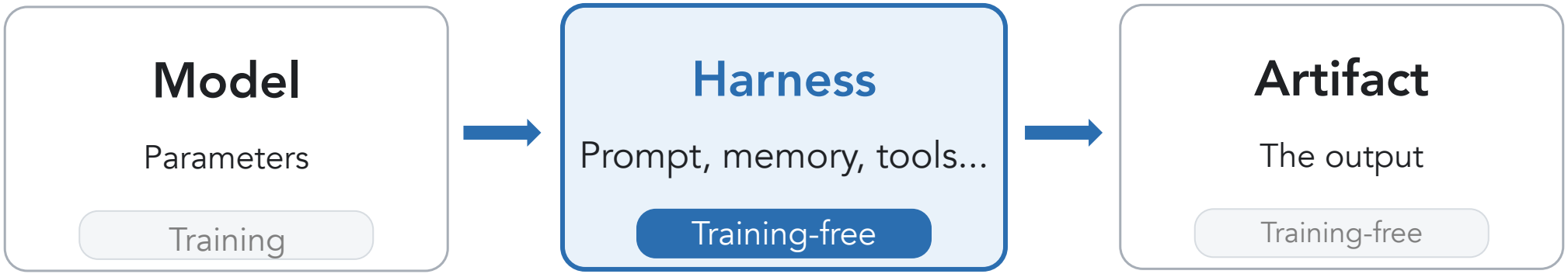
The definition of Recursive Self-Improvement

Ultraintelligent Machine → **Design** is an intellectual activity → **Machine designs better machines** → **Recursive Loop (The Explosion)**

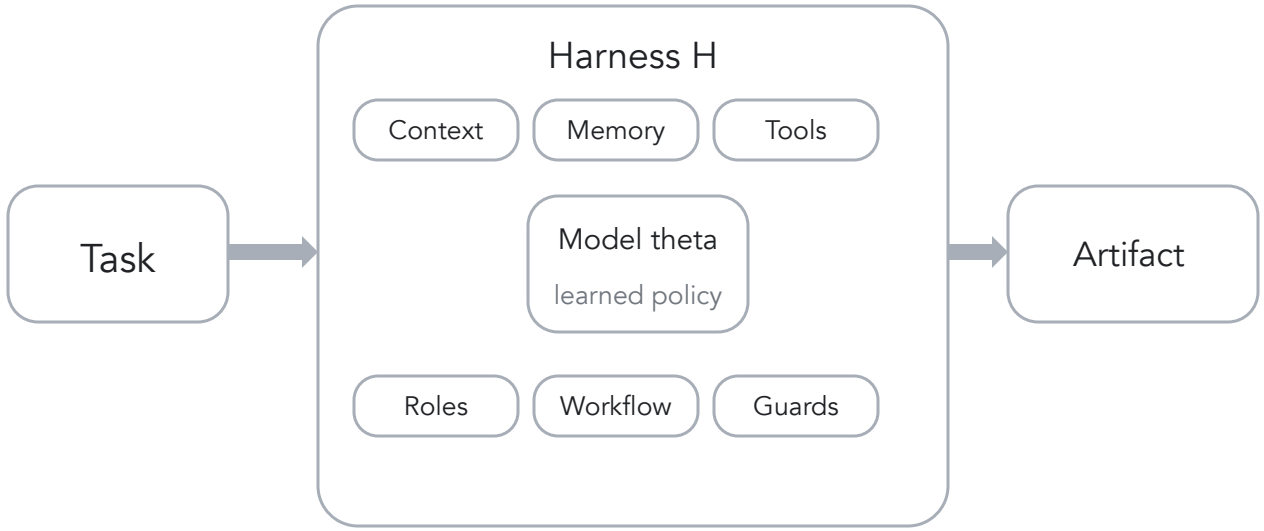
Irving John Good (1965)

The definition of Recursive Self-Improvement

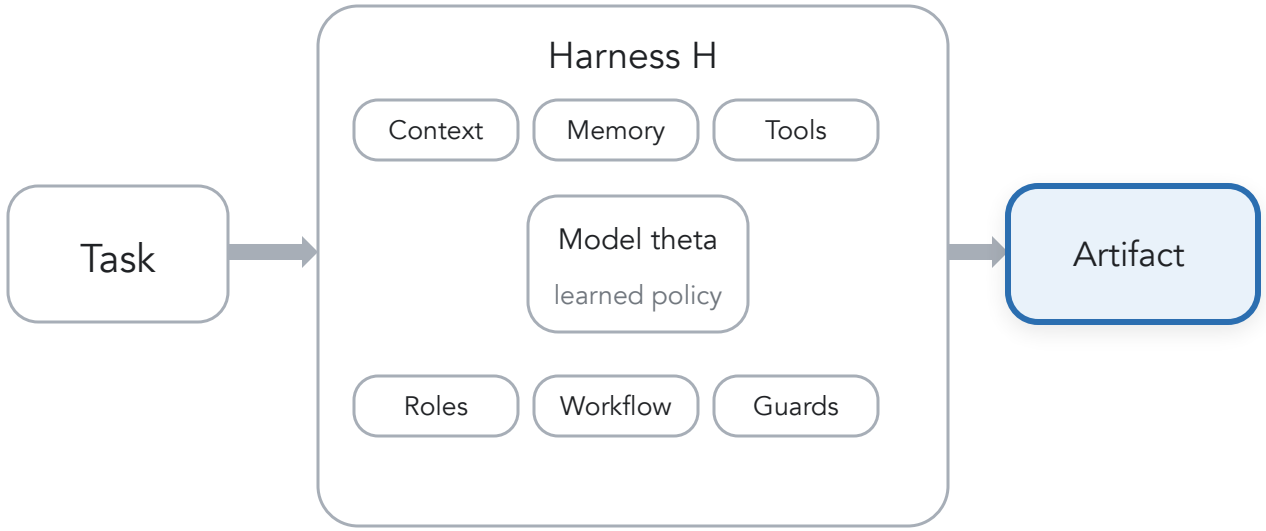
Agent = Model + Harness



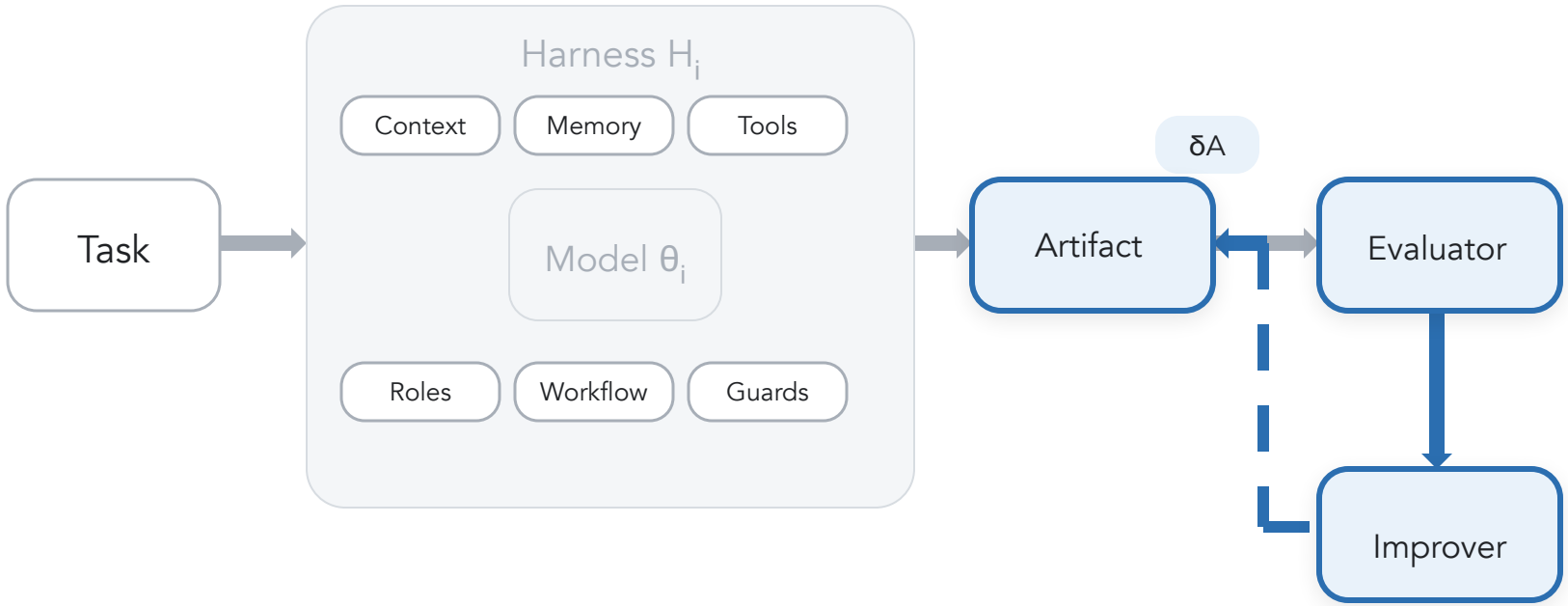
Agent = Model + Harness



Layer 1: Artifact Self-Improvement

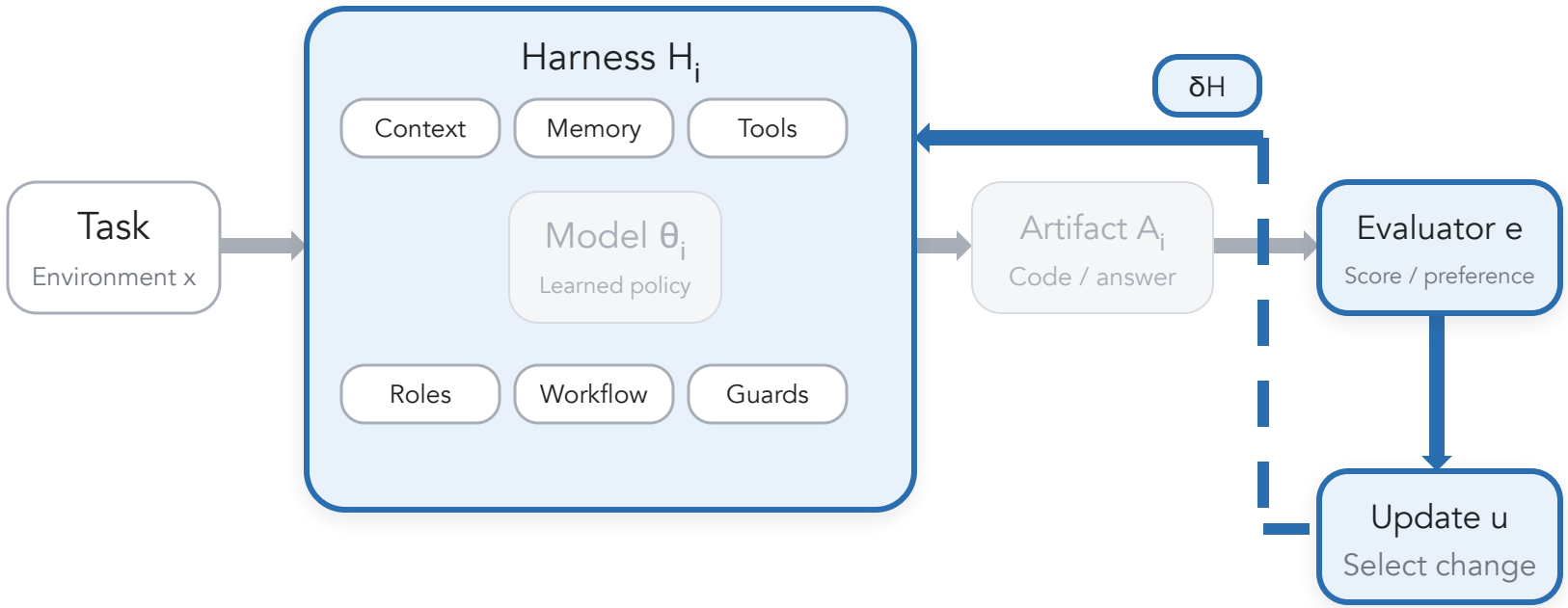


Layer 1: Artifact Self-Improvement



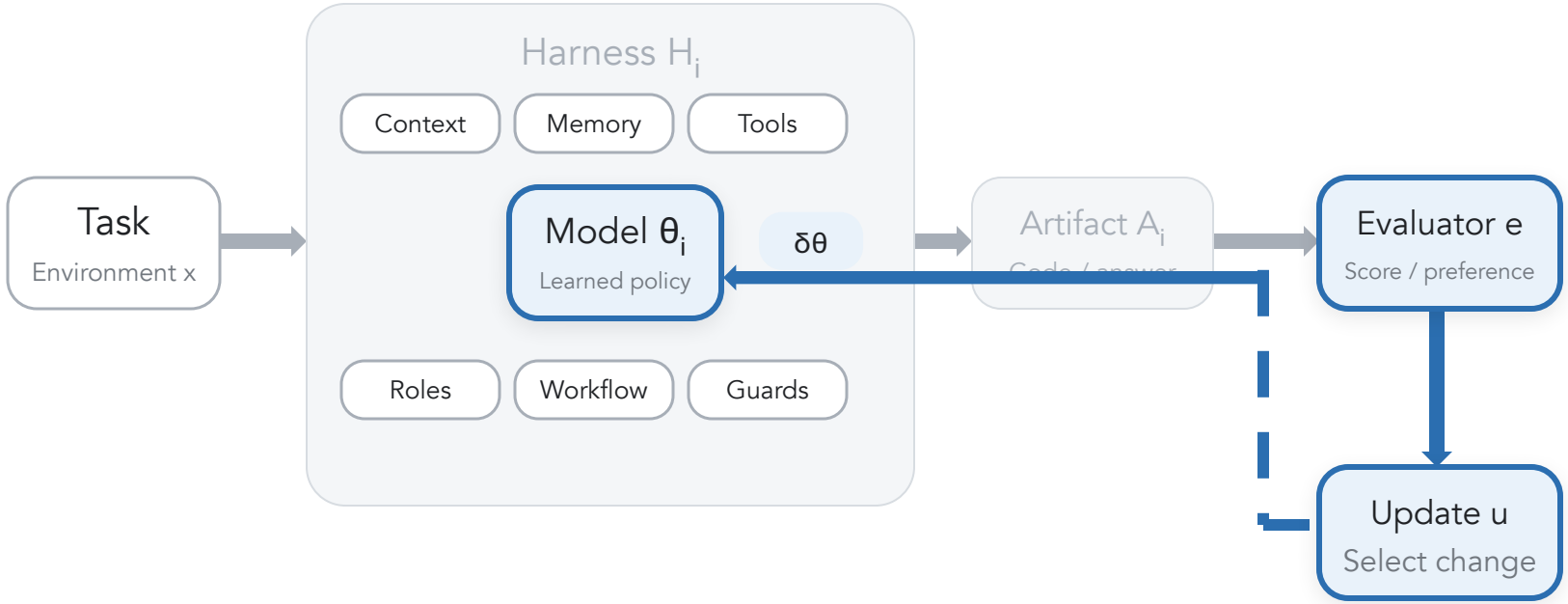
Layer 2: Harness Self-Improvement

Now make the operating system around the model editable

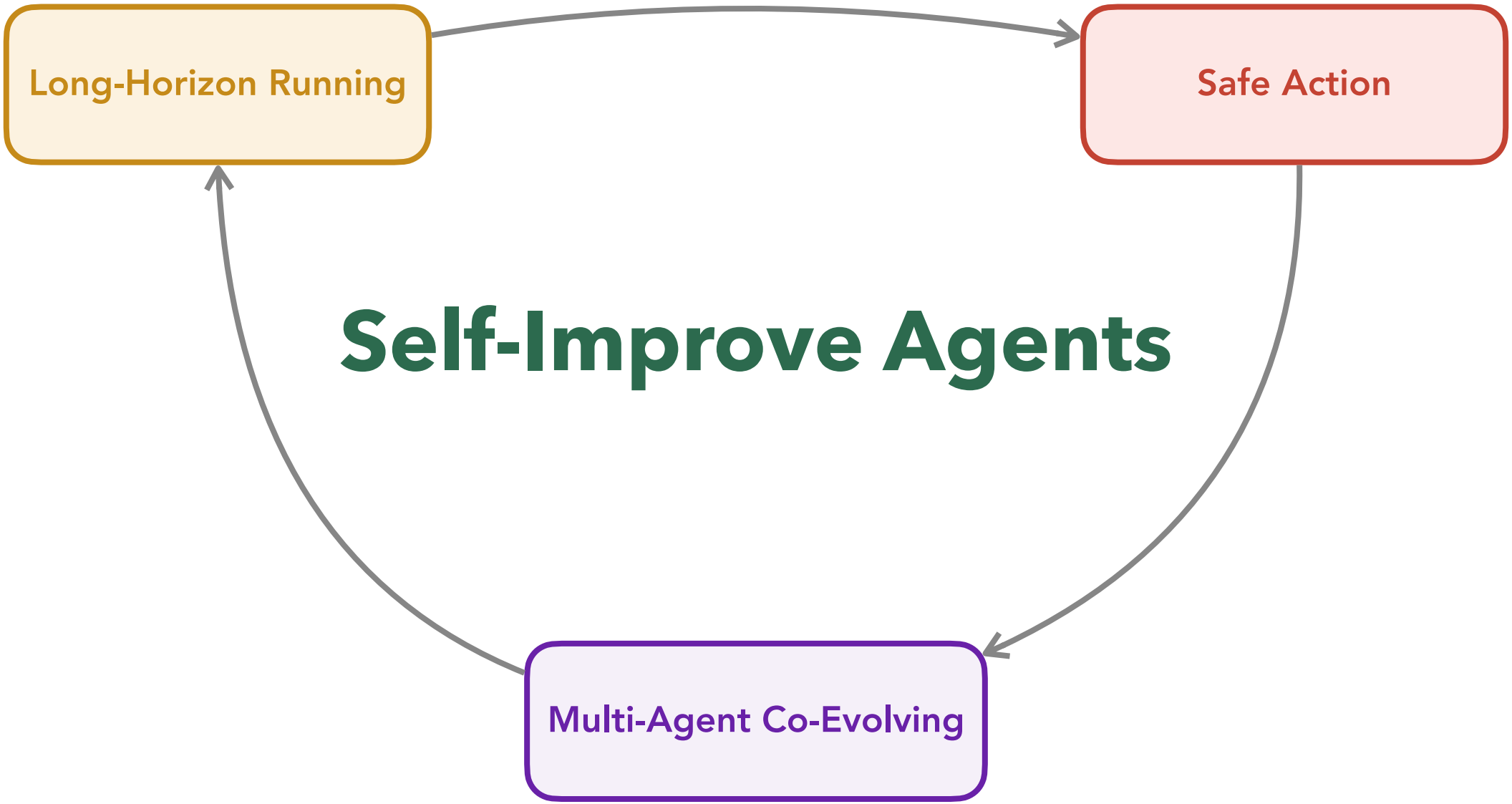


Layer 3: Model Self-Improvement

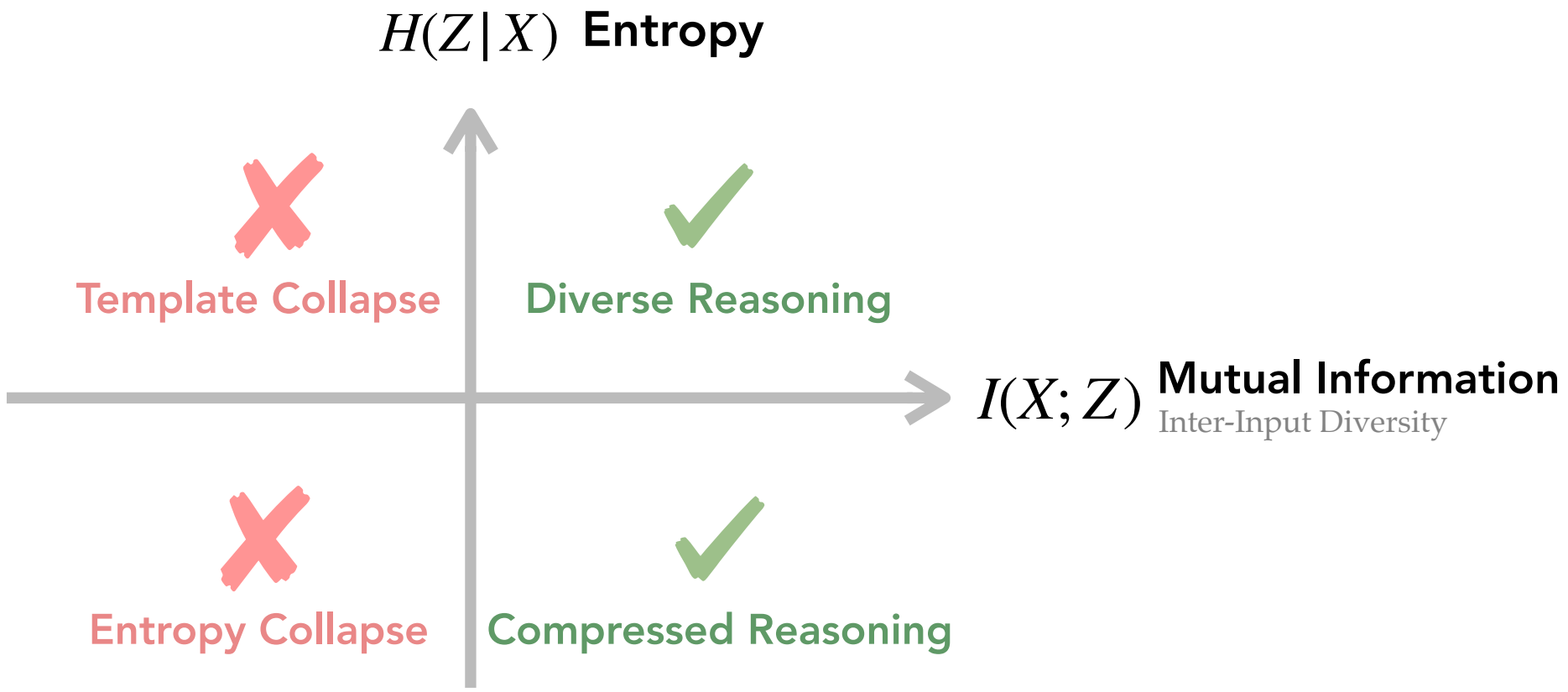
Finally move the write-back path inside the model



Experience is compiled into persistent model parameters.


Long-Horizon Running
Safe Action
Self-Improve Agents
Multi-Agent Co-Evolving

RAGEN-2: Why reasoning collapse in long-horizon?



$$H(Z) = H(Z|X) + I(X;Z)$$

X : *Input*
Z : *Reasoning*

Wang, …, **Li**. RAGEN (2.6K Github Stars, Best Poster Award at Midwest ML 2025)

PRIORS	STAGES	DATA
Common Knowledge	Pre-training	**Static Internet Data:** Books, Images, Videos…
Instruction Following	Self-Supervised Finetuning	**Human Annotations:** <instructions, answers>
World Modeling	Mid-training	**Play Data:** Env, Tools, External Data
Reasoning Planning	Reinforcement Learning	**Task Feedback:** Verifiers (Goal Definitions)
Human Experience	Harness	Prompt Engineering → Context Engineering → Harness Engineering → Graph Engineering

Brute-force Scaling

↓

Continual Learning

(We are here now)

↓

Human + World

(Our future)

PRIORS	STAGES	DATA
Common Knowledge	Pre-training	**Static Internet Data:** Books, Images, Videos…
Instruction Following	Self-Supervised Finetuning	**Human Annotations:** <instructions, answers>
World Modeling	Mid-training	**Play Data:** Env, Tools, External Data
Reasoning Planning	Reinforcement Learning	**Task Feedback:** Verifiers (Goal Definitions)
Human Experience	Harness	**Prompt Engineering** → **Context Engineering** → **Harness Engineering** → **Graph Engineering**

Brute-force Scaling

↓

Continual Learning

(We are here now)

↓

Human + World

(Our future)

world model reasoning through RL

1. What are failure modes of RL?

→ *Exploration vs Exploitation*

2. What priors should we encourage for exploration?

→ *World Model*

3. When is the best stage to encourage it?

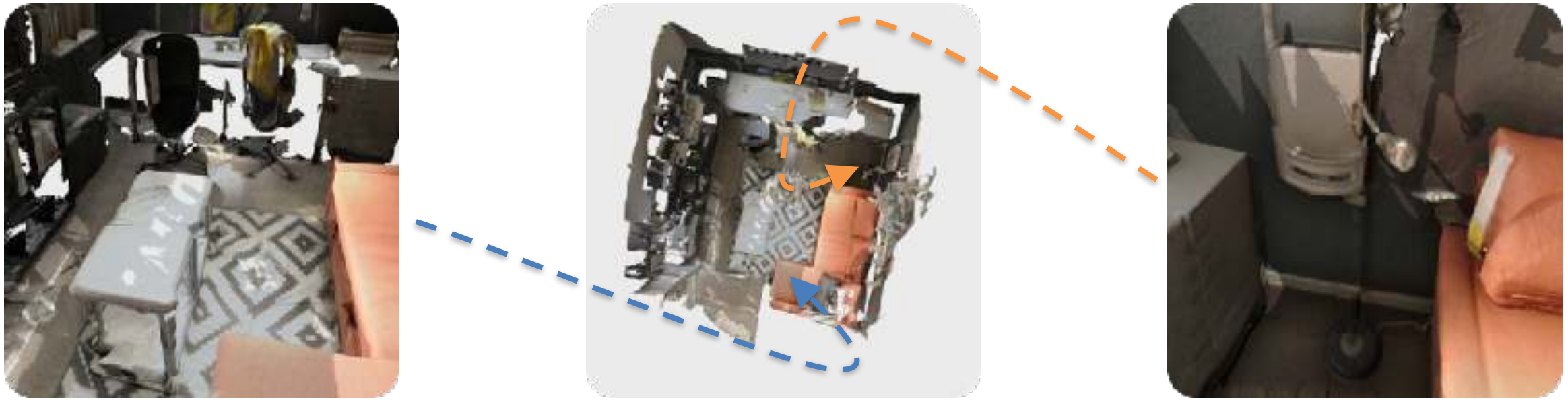
→ *Mid-training vs RL*

Chen, …, **Li**. Self-Play (arxiv 2025)

Planning with the Views

Kangrui Wang, Linjie Li, Zhengyuan Yang, Shiqi Chen, Zihan Wang

Li Fei-Fei, Jiajun Wu, Leonidas Guibas, Lijuan Wang, Manling Li



Initial view

Top-down View

Target view

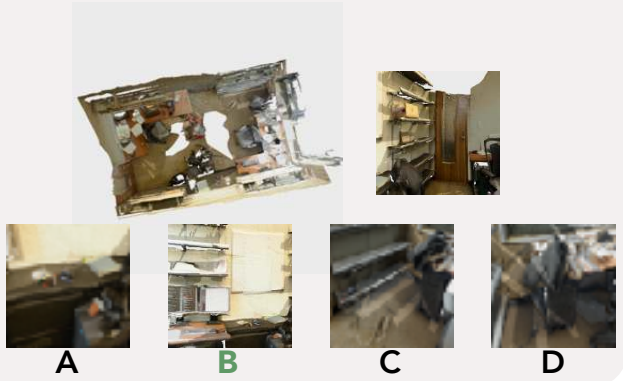
Planning with the Views

Environment. Top: point-cloud scene top-down view with two viewpoints (initial in blue, target in orange). Bottom: rendered first-person views from each viewpoint.

Rendered Initial View

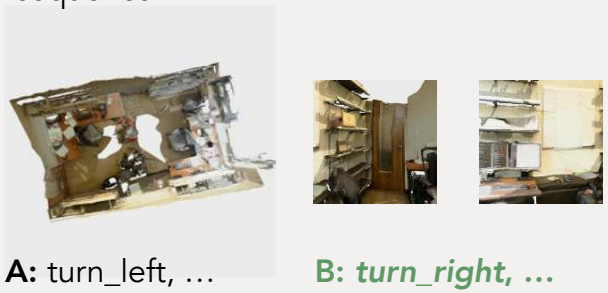
Rendered Target View

Path-to-View. Given top-down, initial view and the action sequence: [turn_right x2, …] , which is the correct target view?



A B C D

View-to-Path. Given top-down, initial and the target view, which is the correct action sequence?



A: turn_left, …

B: *turn_right, …*

C: look_up, …

D: move_left, …

Interactive View Planning. Given top-down, initial and the target view, please find the correct viewpoint of the target view.

Answer(tx, ty, tz, rx, ry, rz)

Model	Path-to-View (P2V)			View-to-Path (V2P)			View Planning (IVP)			Overall
	Short	Long	All	Short	Long	All	Short	Long	All	
Random Response	20.7	24.6	23.3	24.3	26.5	25.7	2.2	0.0	0.8	16.6
Proprietary Models										
GPT-5.4 Pro	**70.7**	**43.8**	**53.1**	**72.4**	39.0	**50.7**	32.6	11.0	18.5	**40.8**
Gemini 3.1 Pro	63.6	40.9	48.8	53.0	**47.7**	49.5	28.8	**17.4**	**21.4**	39.9
GPT-5.4	57.1	42.9	47.8	60.5	37.5	45.6	**33.7**	7.5	16.6	36.7
Grok 4.20 Beta	61.4	38.0	46.1	44.9	44.5	44.6	17.4	2.9	7.9	32.9
GPT-5.1	60.3	35.1	43.9	52.4	33.4	40.1	12.0	3.2	6.2	30.1
Claude Opus 4.6	46.7	28.4	34.8	47.6	38.4	41.6	23.9	3.8	10.8	29.0
Gemini 3 Pro	50.5	31.0	37.8	44.9	35.5	38.8	13.6	7.0	9.3	28.6
Open-Weight Models										
Qwen3.5-397B	**57.6**	30.1	**39.7**	**44.3**	**30.8**	**35.5**	**12.5**	0.0	**4.3**	**26.5**
GLM-4.6V	36.4	23.2	27.8	31.4	29.7	30.2	9.2	**1.2**	4.0	20.7
Qwen2.5-VL-72B	28.3	29.3	28.9	35.7	29.9	31.9	2.2	0.6	1.1	20.7
Qwen3-VL-32B	27.2	27.5	27.4	41.1	28.5	32.9	4.3	0.0	1.5	20.6
Kimi K2.5	35.9	24.6	28.5	18.4	29.4	25.5	4.9	**1.2**	2.5	18.8
Qwen2.5-VL-7B	23.9	**32.5**	29.5	27.0	22.7	24.2	7.1	0.0	2.5	18.7

View Transition like a View World Model

view A

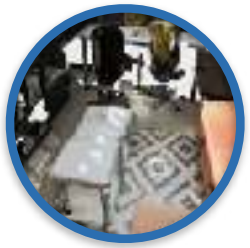
turn left | move forward

view B

Even inside a rollout that missed its goal, going A→B is a view transition that can serve as valid supervision.

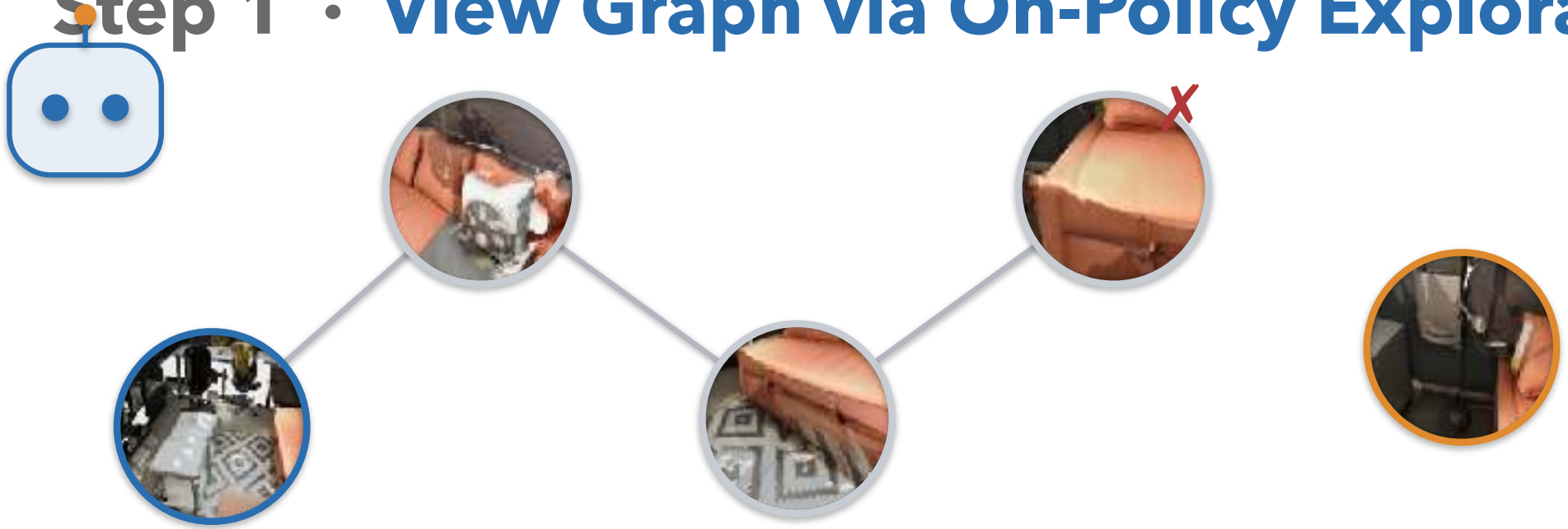
Continual Learning with a View World Model?

Step 1 · View Graph via On-Policy Exploration



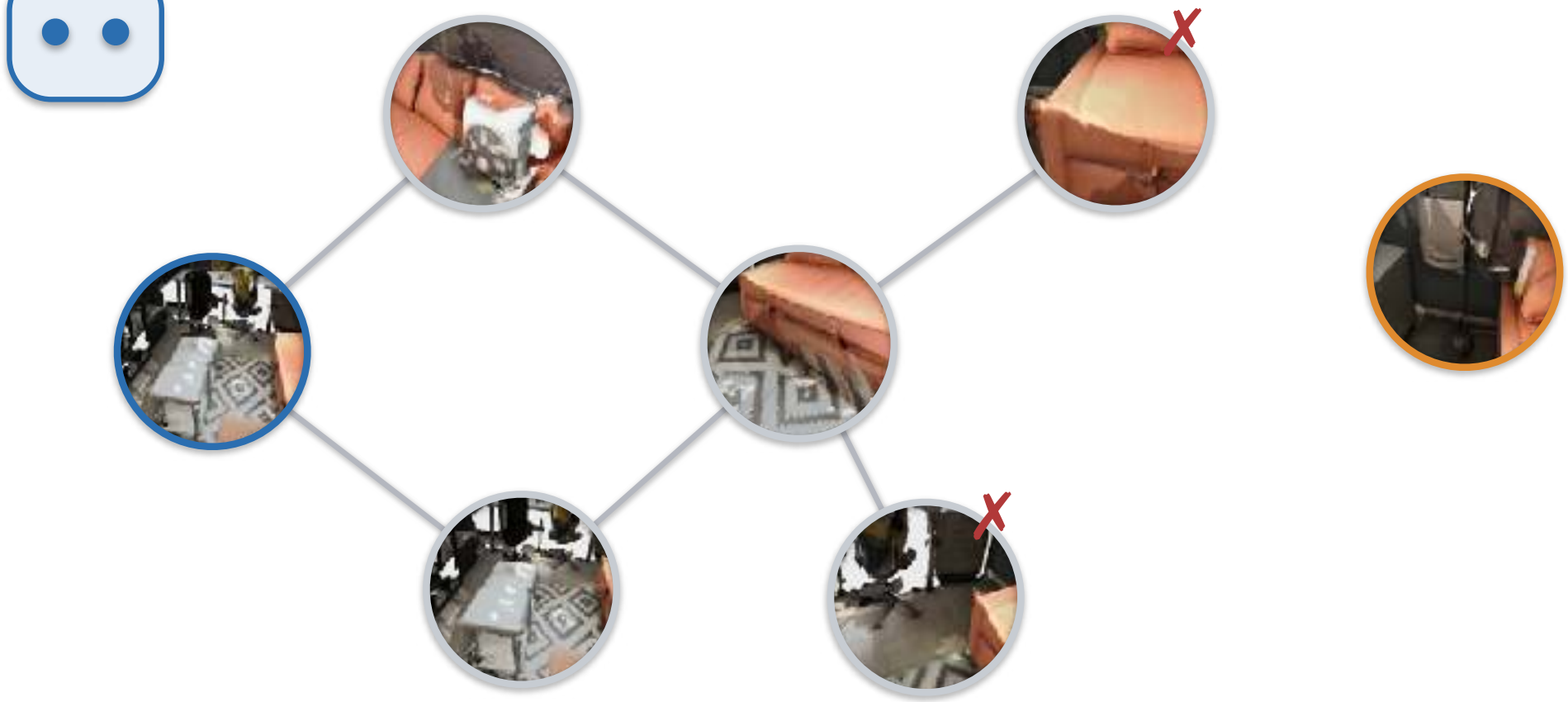
The agent explores ViewSuite scenes from its own experience.

Step 1 · View Graph via On-Policy Exploration



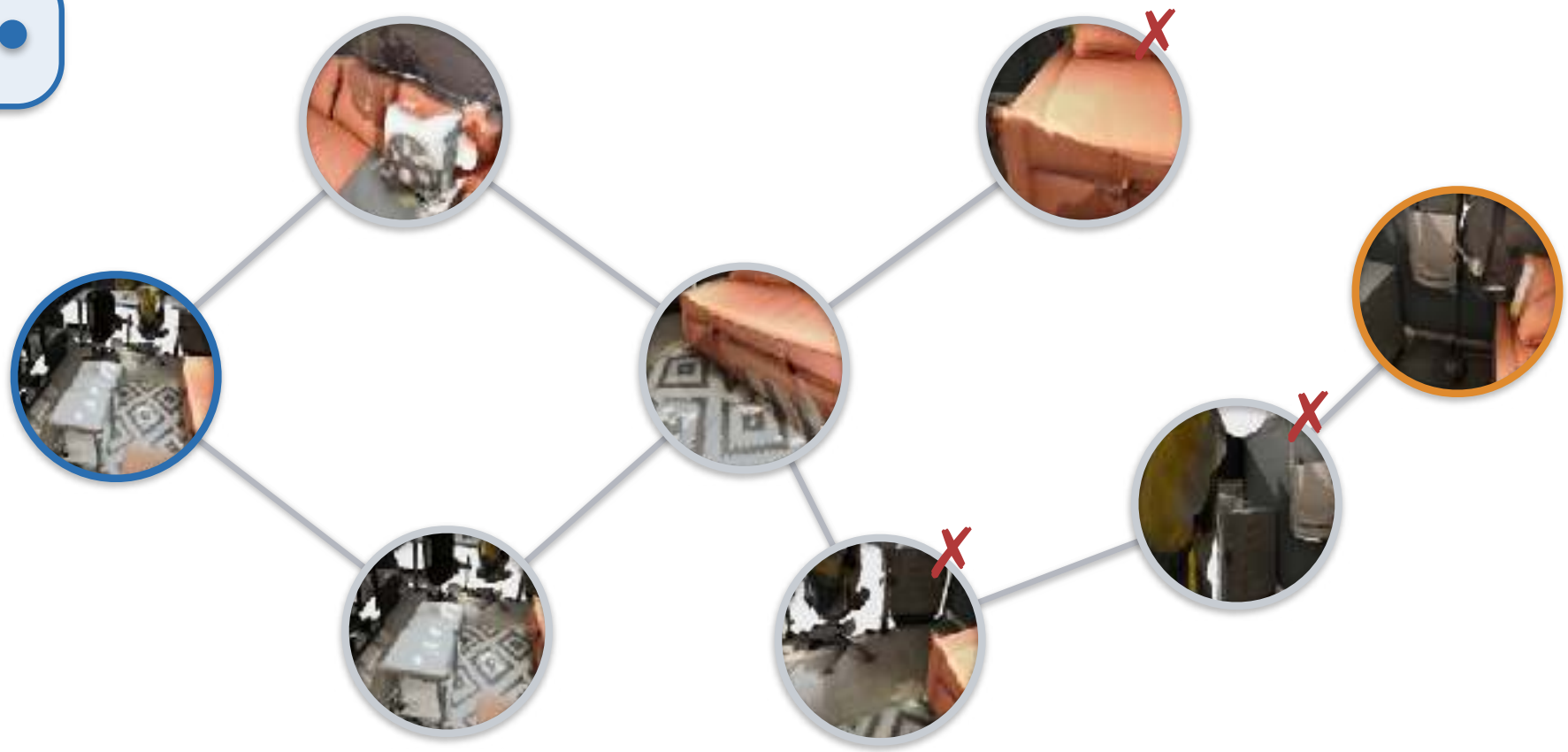
Rollout 1 wanders the scene – and misses.

Step 1 · View Graph via On-Policy Exploration



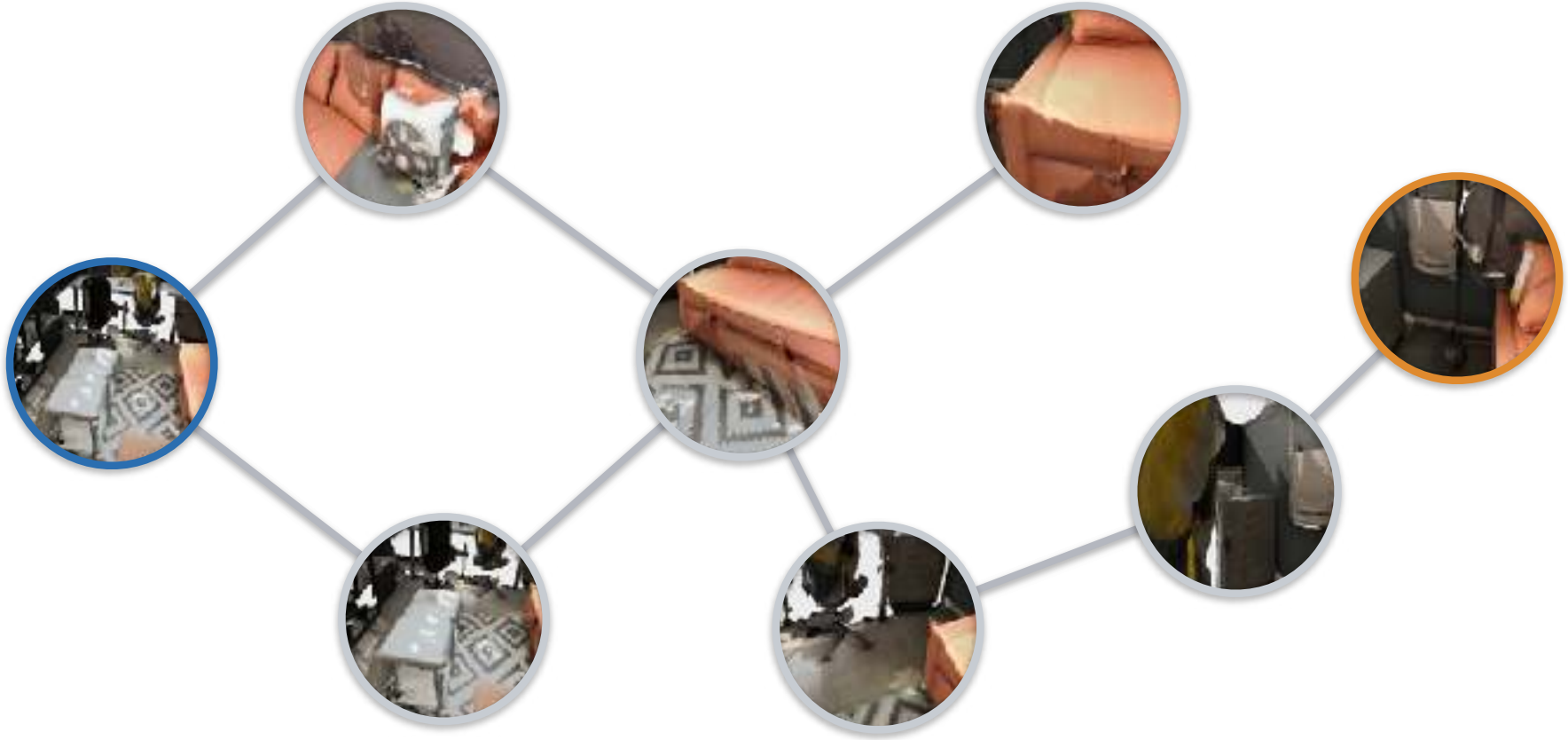
Rollout 2 also fails – but it crosses rollout 1.

Step 1 · View Graph via On-Policy Exploration



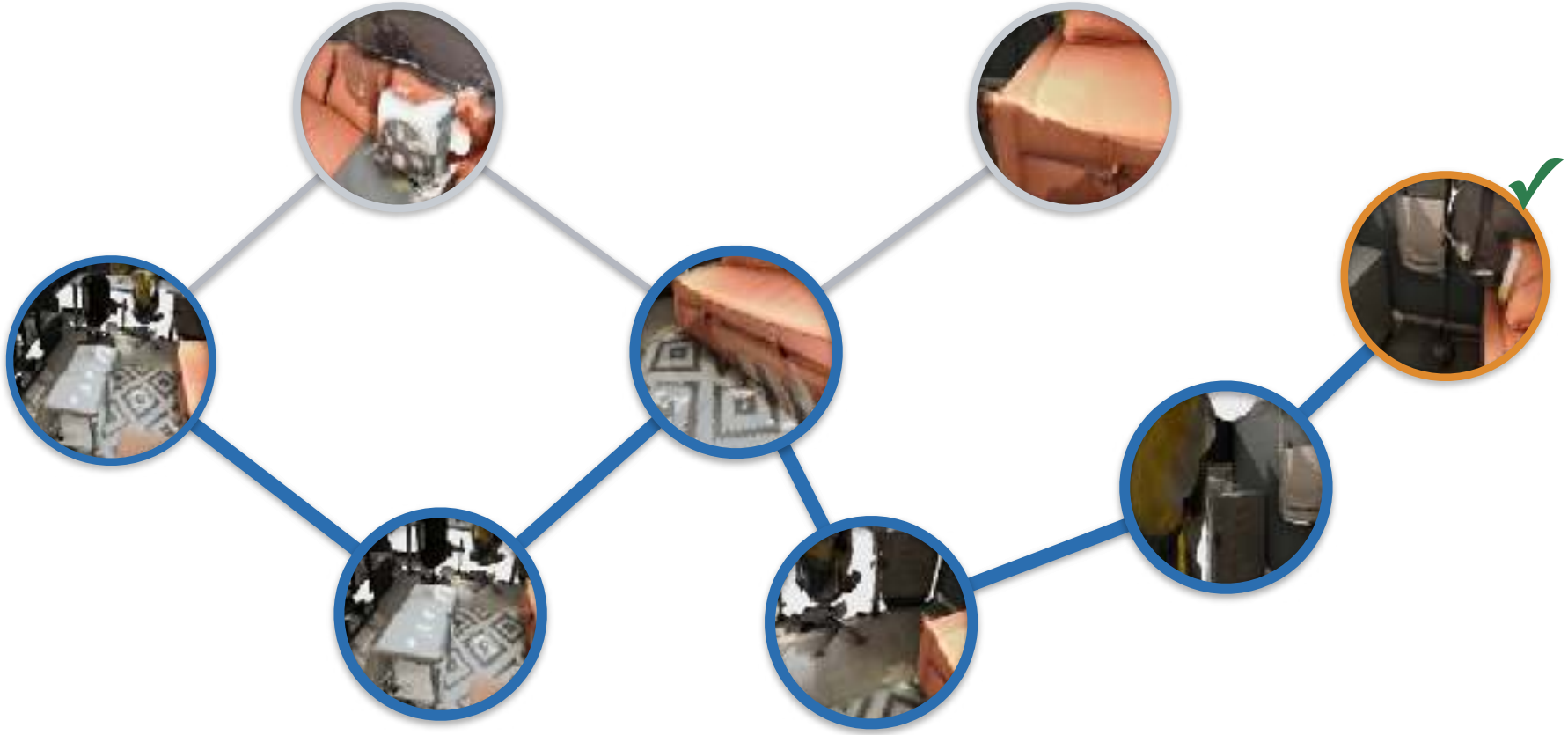
Every rollout misses the target. **Yet together they trace the whole scene.**

Step 2 · A View Graph Emerges



Shared viewpoints merge: nodes are views, edges are actions.

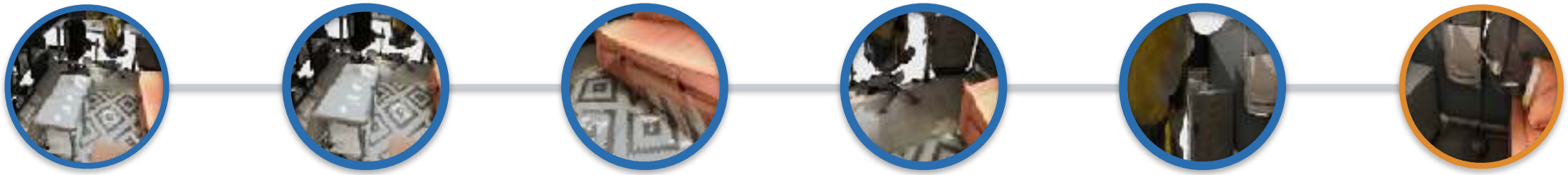
Step 2 · A Path Emerges



A path the agent never walked now connects initial → target.

Step 3 · Reformulate into a Demo

initial view **target view**



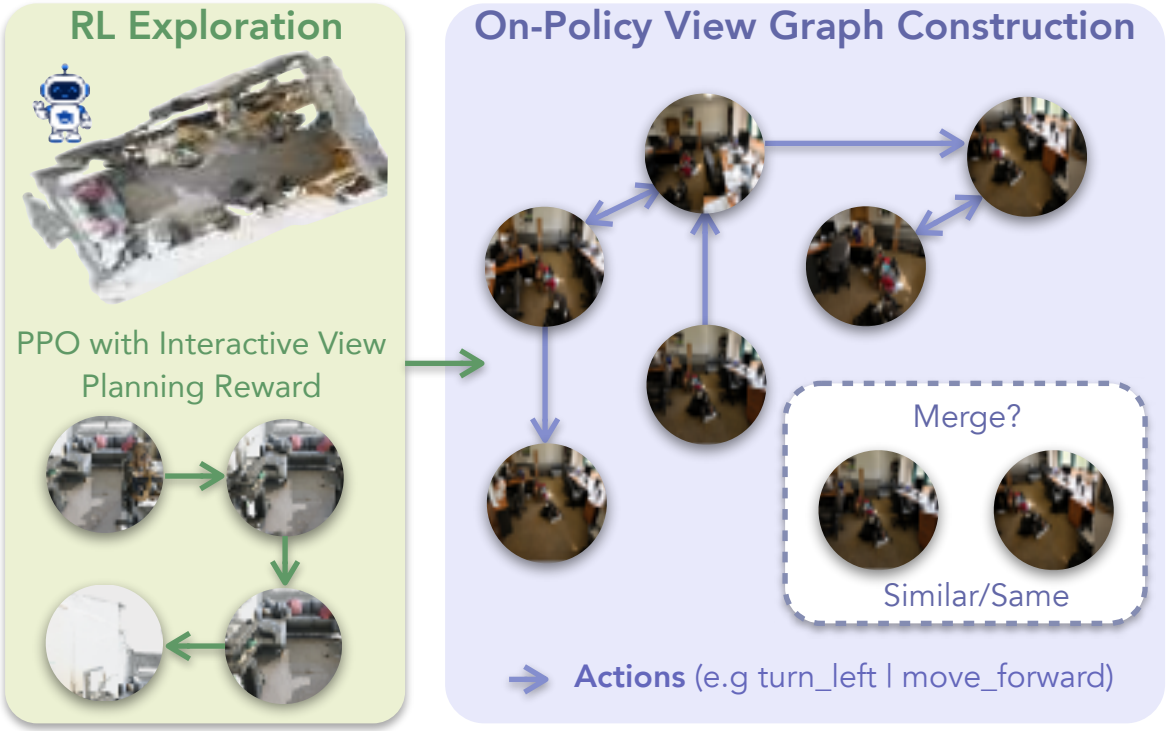
turn L · fwd | move L · up | fwd · turn x2 | turn · look up | up · right

input: initial + target view ***output: the action sequence***

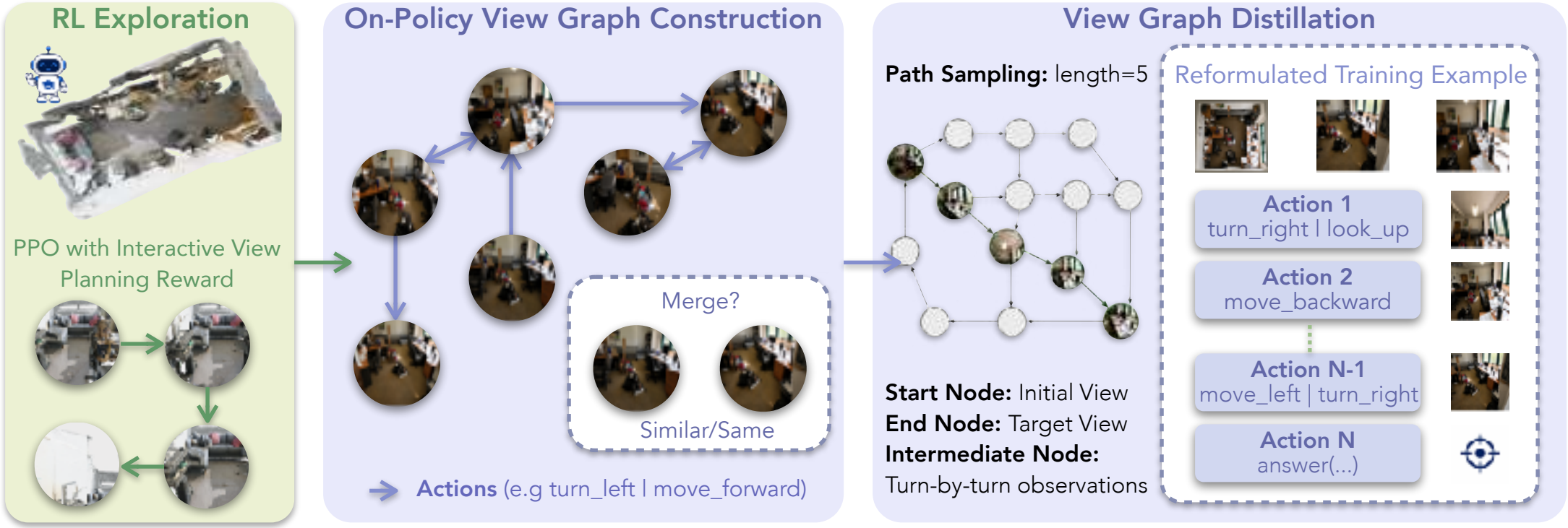
Any sampled path becomes a demo
valid even though no rollout reached the target.

View Planning via Self-Exploration

View Planning via Self-Exploration



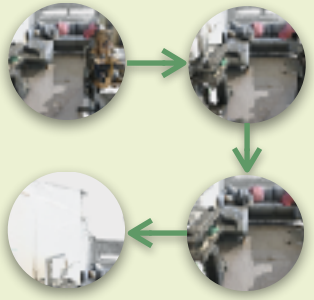
View Planning via Self-Exploration



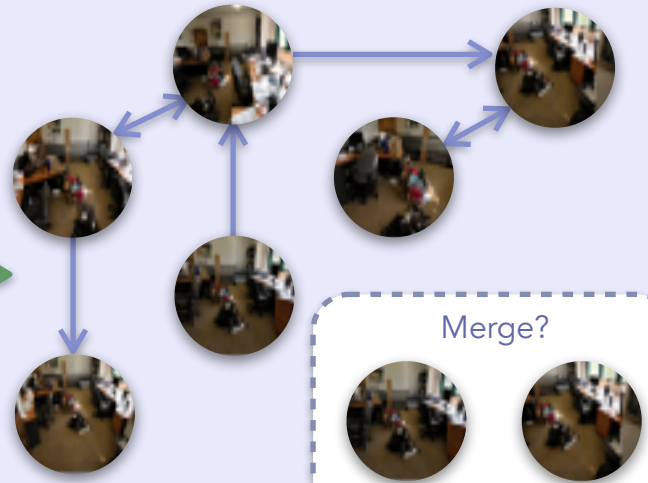
View Planning via Self-Exploration

RL Exploration

PPO with Interactive View Planning Reward



On-Policy View Graph Construction



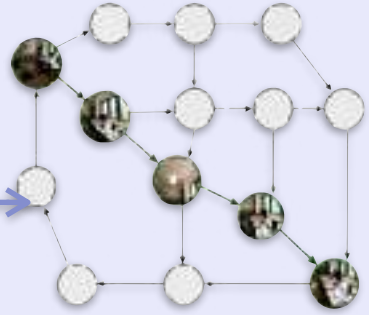
Merge?

Similar/Same

Actions (e.g turn_left | move_forward)

View Graph Distillation

Path Sampling: length=5



Start Node: Initial View
End Node: Target View
Intermediate Node: Turn-by-turn observations

Reformulated Training Example

Action 1
turn_right | look_up

Action 2
move_backward

Action N-1
move_left | turn_right

Action N
answer(...)

SFT Training

Interactive View Planning

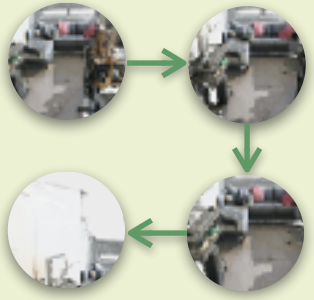
View Difference

Cross-entropy loss on reformulated data

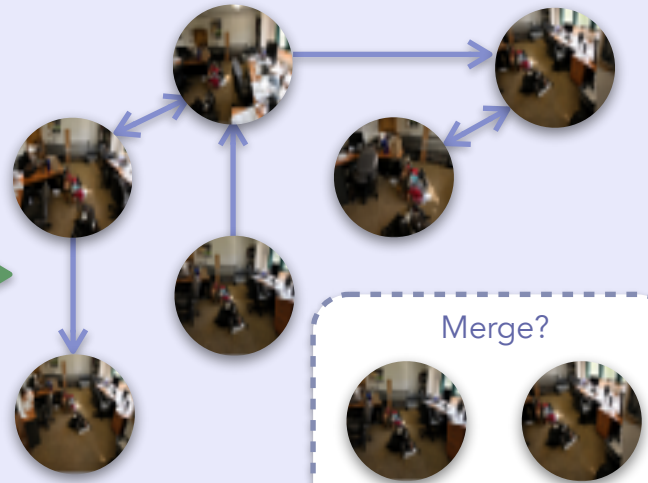
View Planning via Self-Exploration

RL Exploration

PPO with Interactive View Planning Reward



On-Policy View Graph Construction



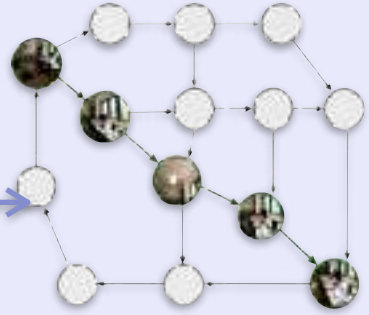
Merge?

Similar/Same

Actions (e.g turn_left | move_forward)

View Graph Distillation

Path Sampling: length=5



Start Node: Initial View
End Node: Target View
Intermediate Node: Turn-by-turn observations

Reformulated Training Example

Action 1
turn_right | look_up

Action 2
move_backward

Action N-1
move_left | turn_right

Action N
answer(...)

SFT Training

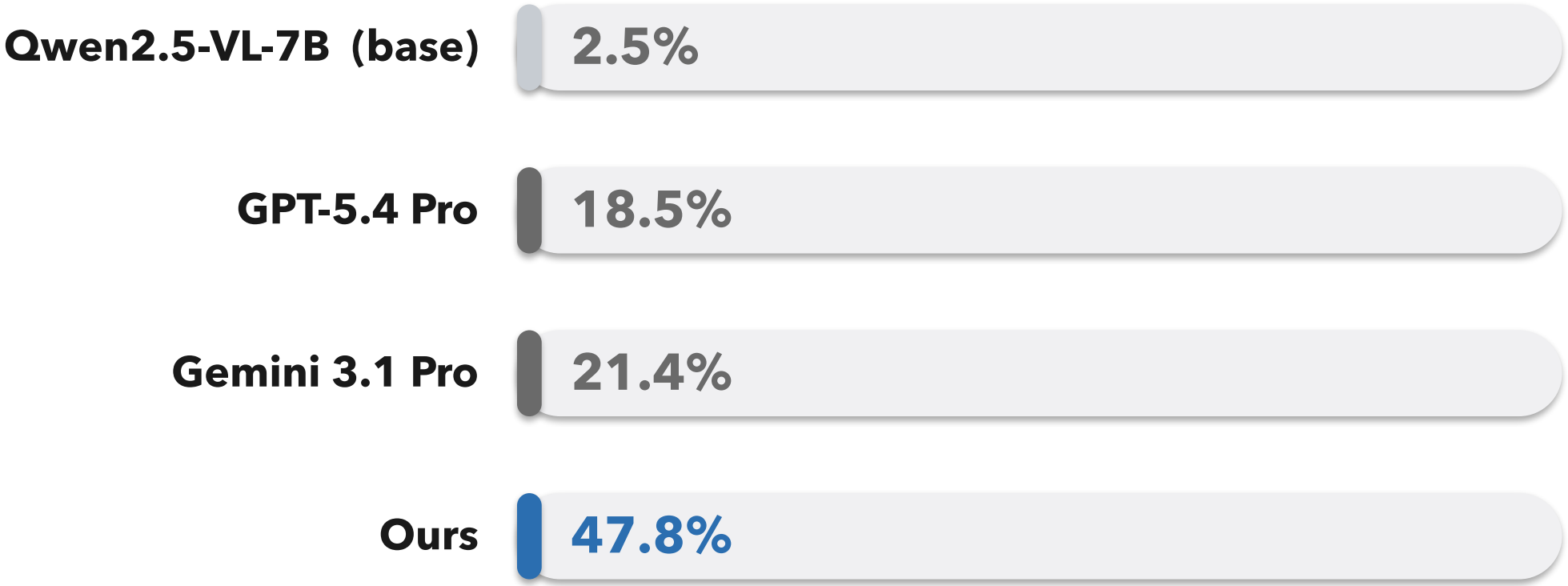
Interactive View Planning

View Difference

Cross-entropy loss on reformulated data

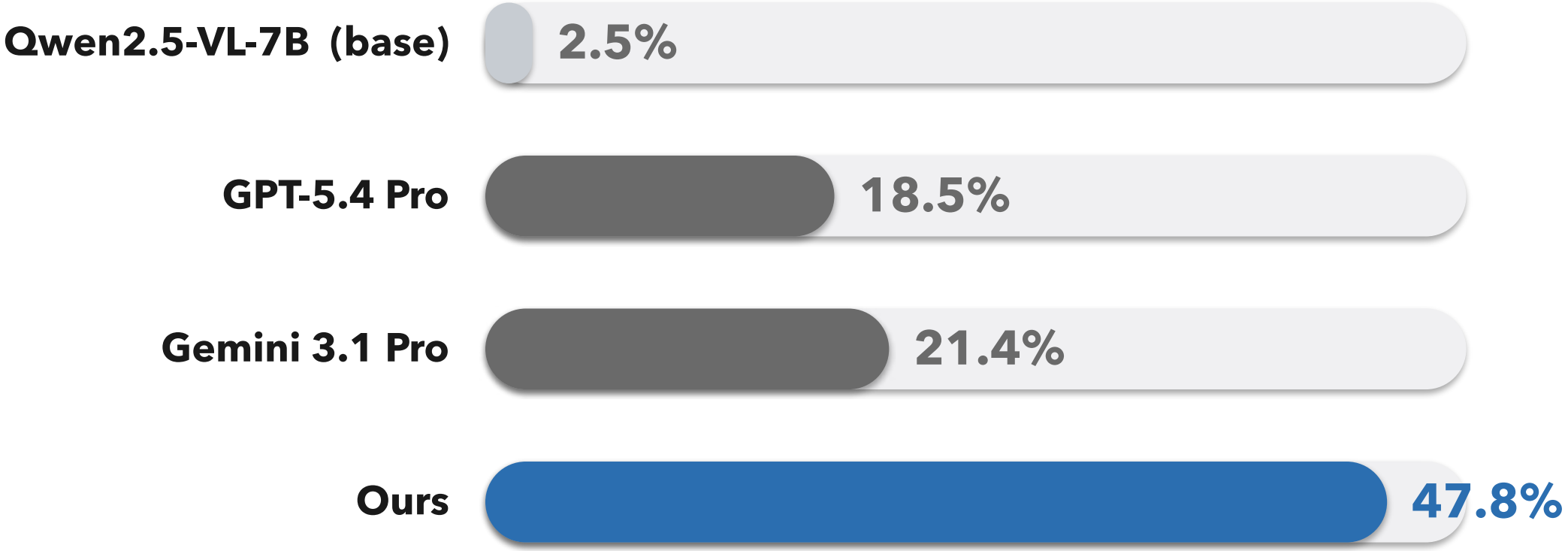
Next Iteration

From 2.5% to 47.8%



View Planning (arxiv 2026)

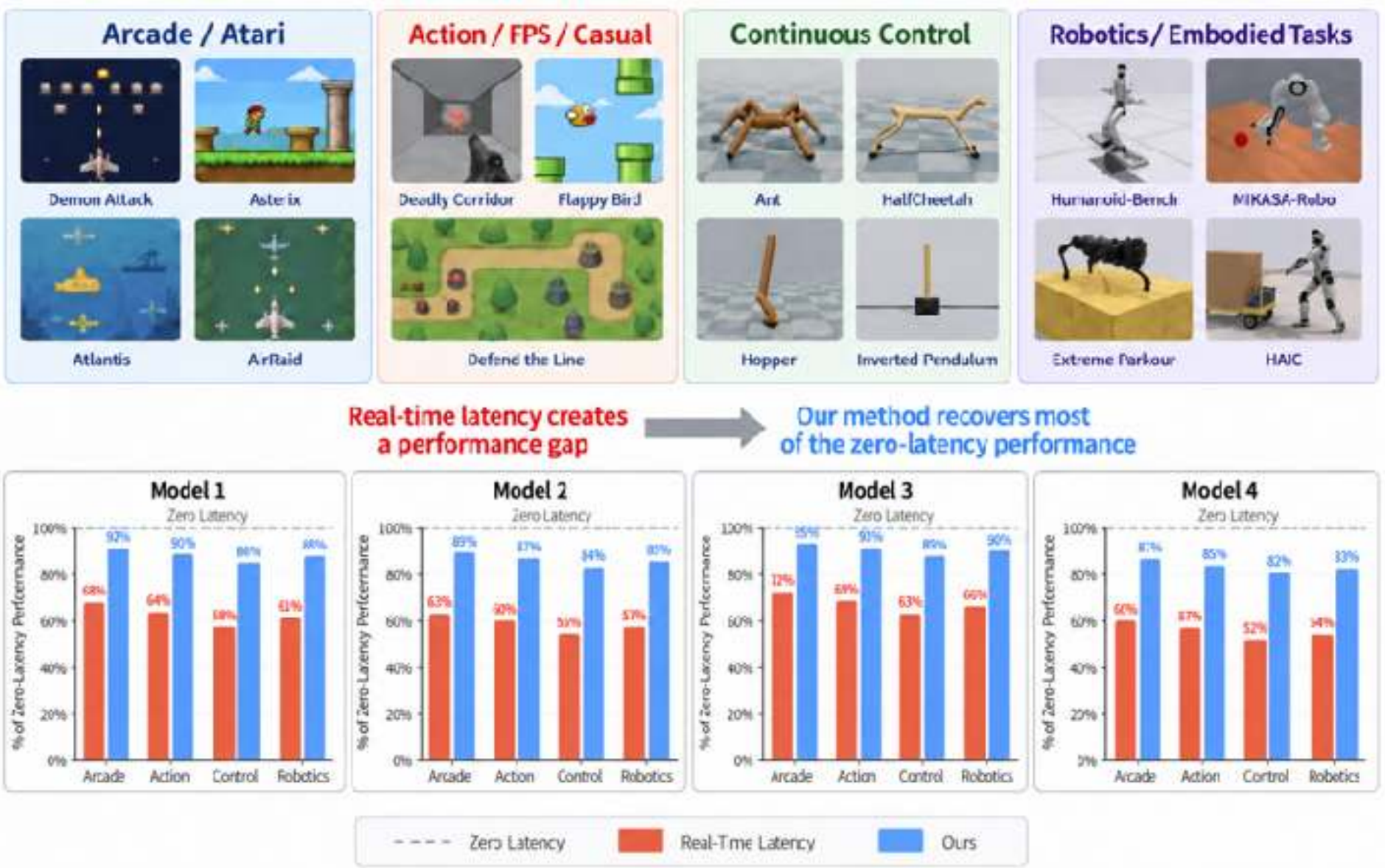
From 2.5% to 47.8%

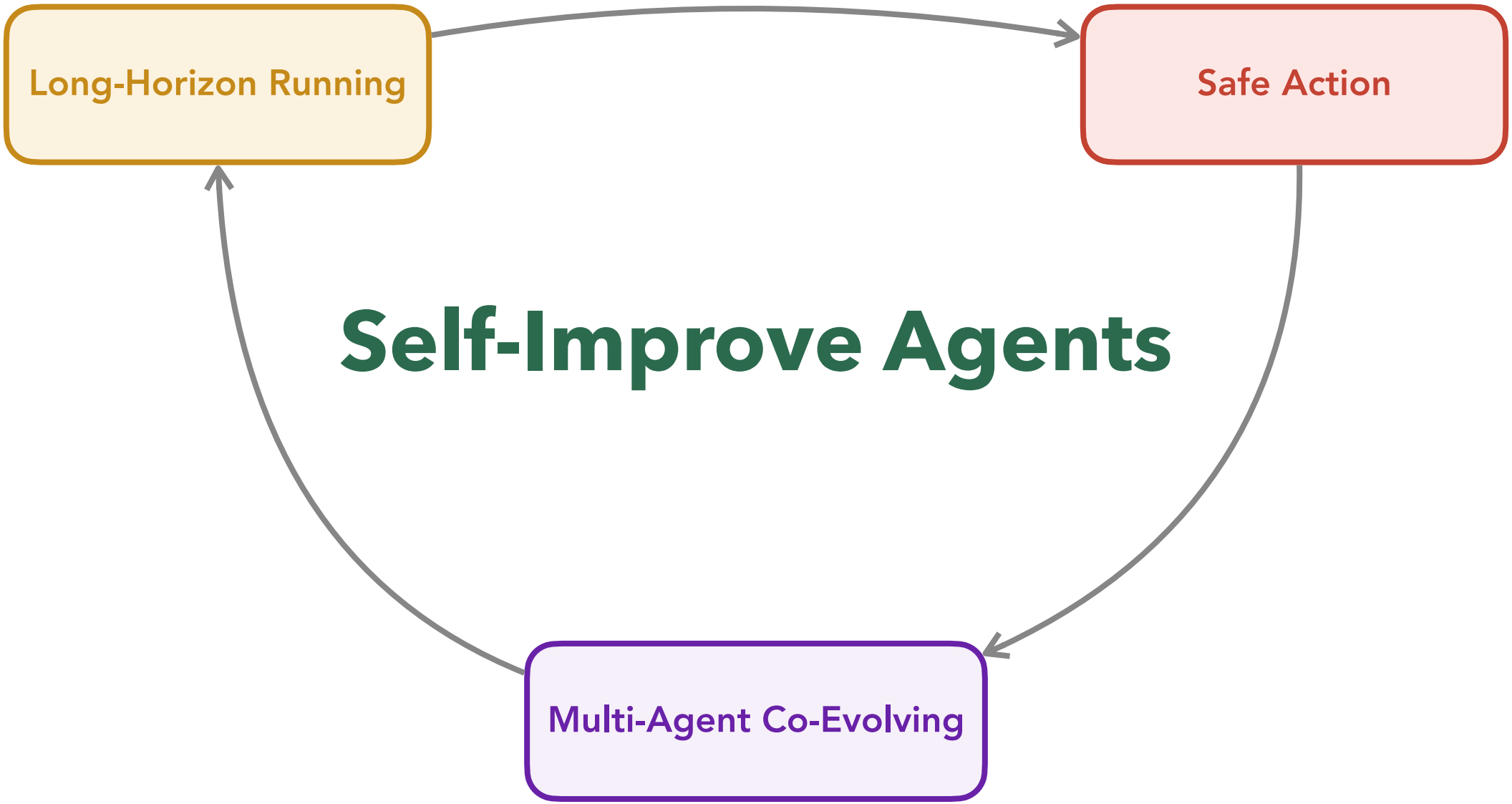


A 7B model trained only on its own exploration beats every frontier VLM.

View Planning (arxiv 2026)

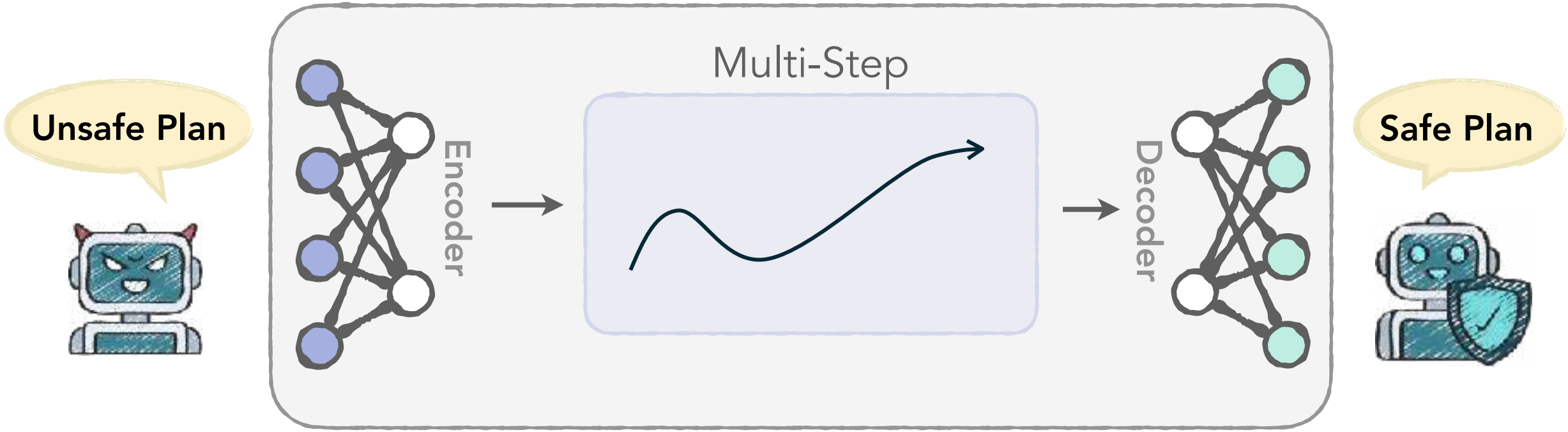
Another Direction: Latency-Aware Planning, Budget-Aware Planning




Long-Horizon Running
Safe Action
Self-Improve Agents
Multi-Agent Co-Evolving

ODE-Steering

Modeling a as ODE

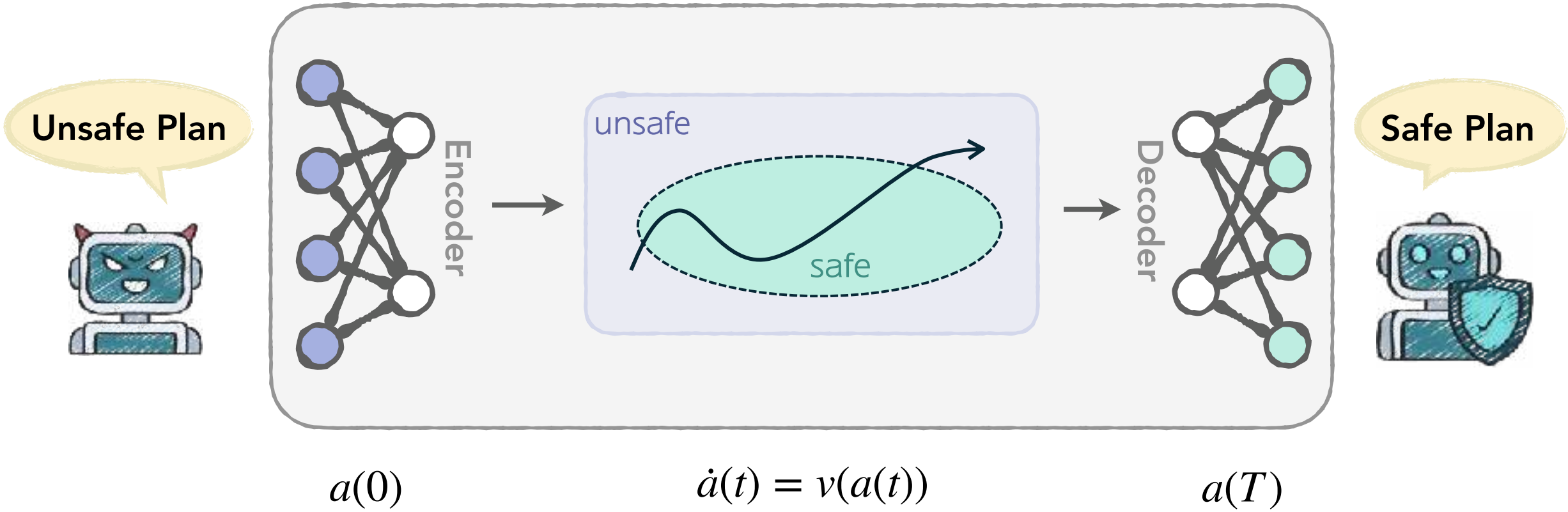


$a(0)$ $\dot{a}(t) = v(a(t))$ $a(T)$

Zhao, …, **Li***, Shao*. ODE-Steer (ICLR 2026)

ODE-Steering: Control Barrier Function

If we can draw a **safe boundary** for a

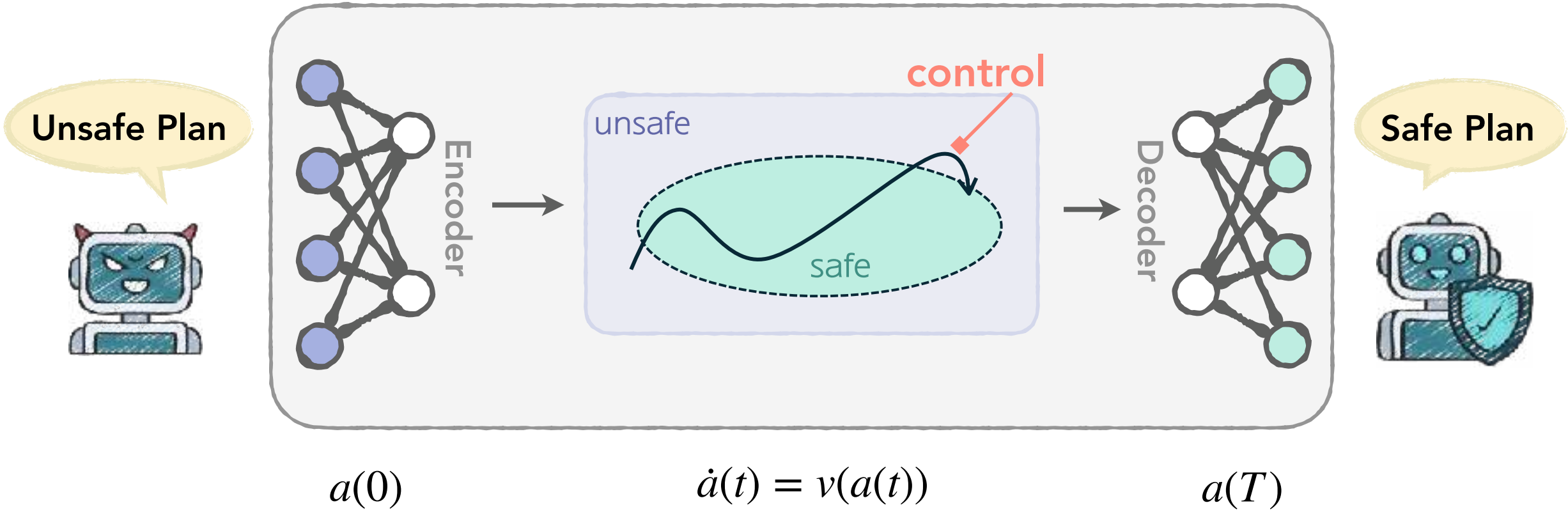


$a(0)$ $\quad \dot{a}(t) = v(a(t)) \quad$ $a(T)$

Zhao, …, **Li***, Shao*. ODE-Steer (ICLR 2026)

ODE-Steering: Control Barrier Function

If we can draw a **safe boundary** for a

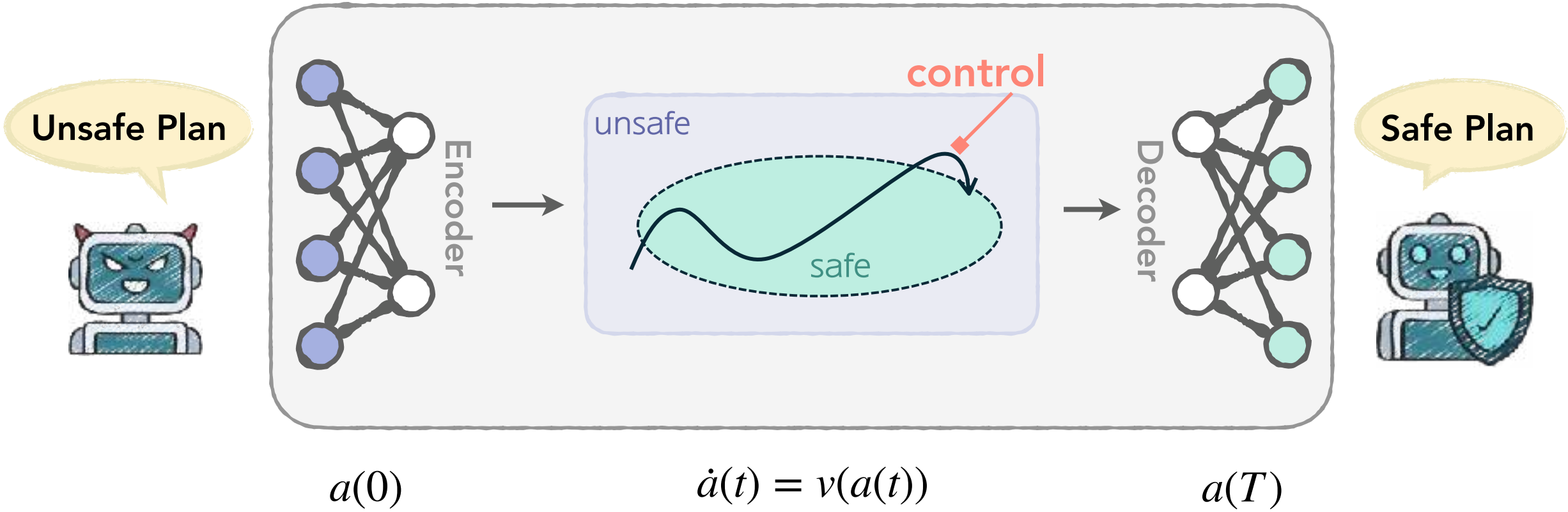


$a(0)$ $\quad \dot{a}(t) = v(a(t)) \quad$ $a(T)$

Zhao, …, **Li***, Shao*. ODE-Steer (ICLR 2026)

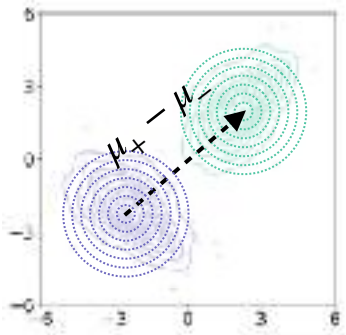
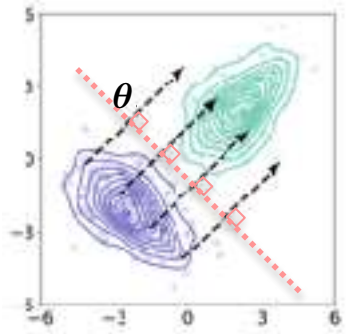
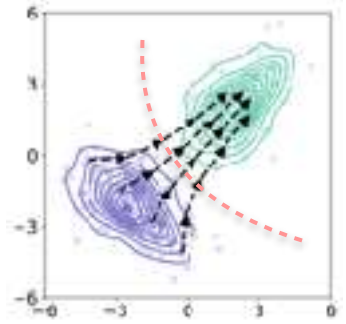
ODE-Steering: Control Barrier Function

How to learn the **safe boundary** h ?

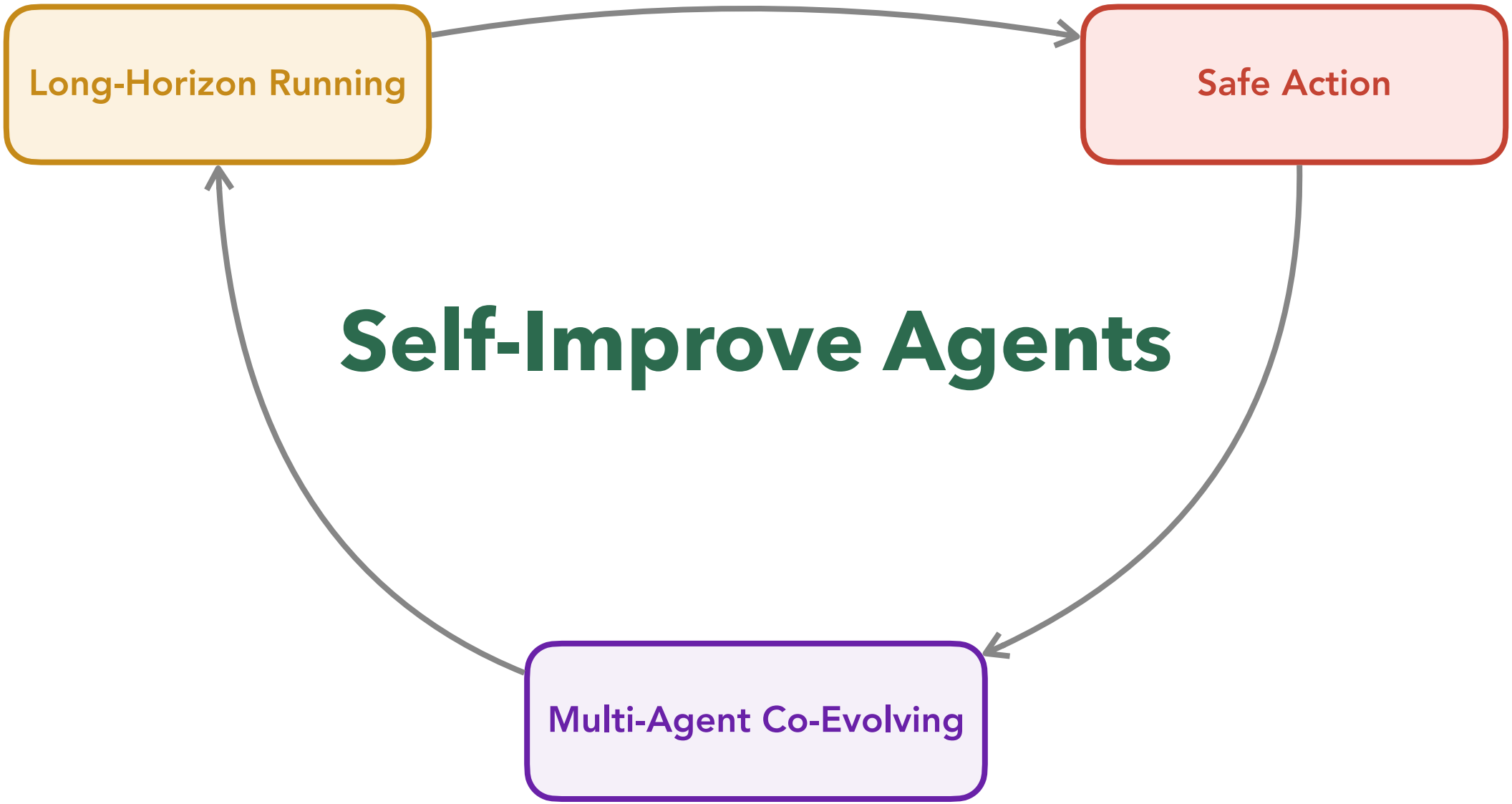


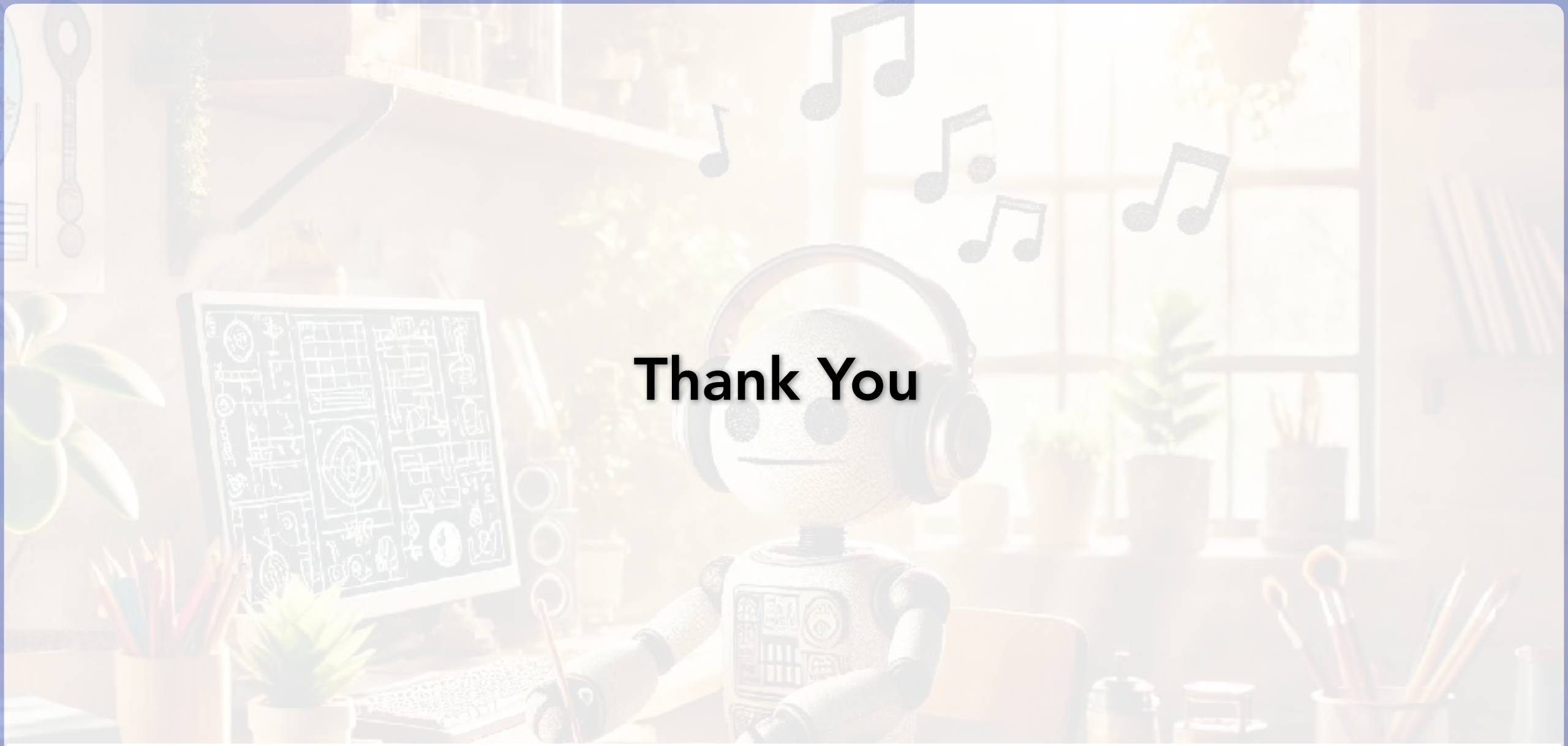
$a(0)$ $\quad \dot{a}(t) = v(a(t)) \quad$ $a(T)$

Zhao, …, **Li***, Shao*. ODE-Steer (ICLR 2026)

Steering Vector $v(a) = \nabla h(a)$	Barrier Function $h(a)$
 Difference-in-Means [1] $\mu_+ - \mu_-$	Log-density ratio $h(\boldsymbol{a}) = \log \frac{p_+(\boldsymbol{a})}{p_-(\boldsymbol{a})}$ Gaussian Assumption $p_\pm(\boldsymbol{a}) = \mathcal{N}(\boldsymbol{\mu}_\pm, \boldsymbol{I})$
 Linear Probe [2] $\boldsymbol{\theta}$	Log-density ratio $h(\boldsymbol{a}) = \log \frac{p_+(\boldsymbol{a})}{p_-(\boldsymbol{a})}$ Logistic Regression $p_{\boldsymbol{\theta}}(\boldsymbol{a}) = \text{sigmoid}(\boldsymbol{\theta}^\top \boldsymbol{a})$.
 $\nabla h(\boldsymbol{a}) = \boldsymbol{J}_{\boldsymbol{\phi}}(\boldsymbol{a})^\top \boldsymbol{w}$	Log-density ratio $h(\boldsymbol{a}) = \log \frac{p_+(\boldsymbol{a})}{p_-(\boldsymbol{a})}$ $= \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{a}) + b$ nonlinear feature map

Zhao, …, **Li***, Shao*. ODE-Steer (ICLR 2026)


Long-Horizon Running
Safe Action
Self-Improve Agents
Multi-Agent Co-Evolving



MLL LAB
Machine Learning and Language