Lecture 21 - Reinforcement Learning in Embodied AI

Guiliang Liu

The Chinese University of Hong Kong, Shenzhen

DDA4230: Reinforcement Learning
Course Page: [Click]

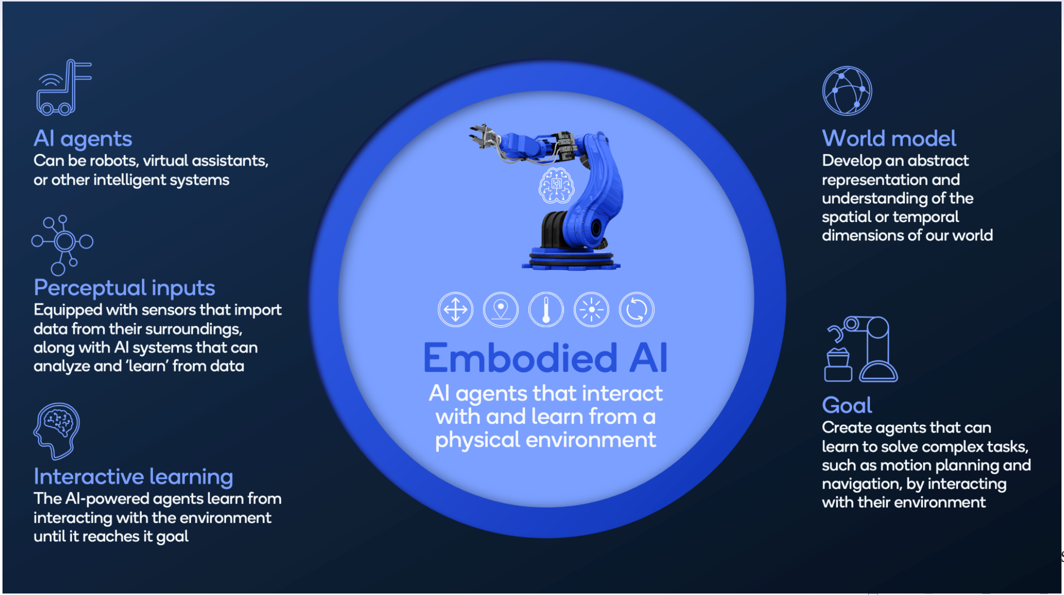
An Introduction to Embodied AI

What is "Embodied AI"?

Embodied $\rightarrow$ "Possessing or existing in bodily form".

Embodied AI learns through interactions with environments from an egocentric perception similar to humans, instead of learning from a fixed dataset.

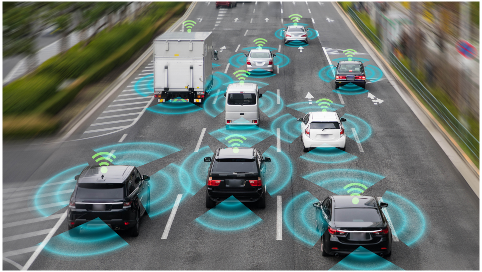
- **Data-Driven AI:** Learning from a fixed demonstration dataset.
- **Embodied AI:** Learning by interacting with the environment.

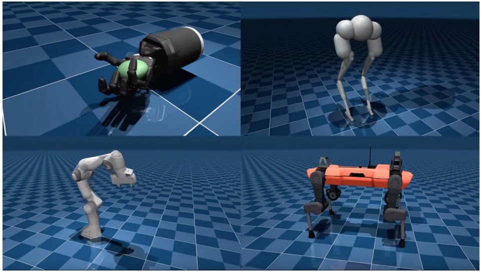

AI agents
Can be robots, virtual assistants, or other intelligent systems
Perceptual inputs
Equipped with sensors that import data from their surroundings, along with AI systems that can analyze and 'learn' from data
Interactive learning
The AI-powered agents learn from interacting with the environment until it reaches it goal
Embodied AI
AI agents that interact with and learn from a physical environment
World model
Develop an abstract representation and understanding of the spatial or temporal dimensions of our world
Goal
Create agents that can learn to solve complex tasks, such as motion planning and navigation, by interacting with their environment

An Introduction to Embodied AI

Example of embodied AI:

Autonomous Driving (e.g., SUMO, Carla). Robot Control (e.g., MuJoCo, Issac Gym).

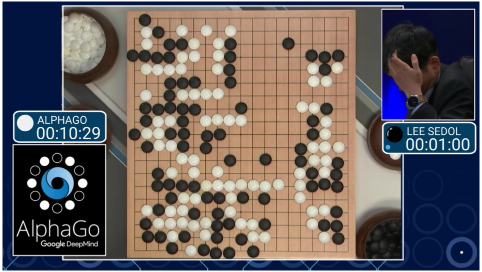




An Introduction to Embodied AI

Example of embodied AI:

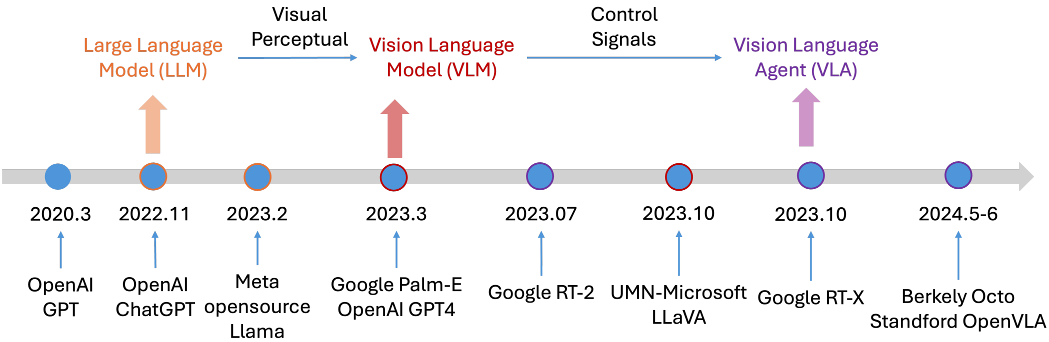
Board Games (e.g., AlphaGo, AlphaZero). Video Games (e.g., AlphaStar, OpenAI5).



香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

An Introduction to Embodied AI

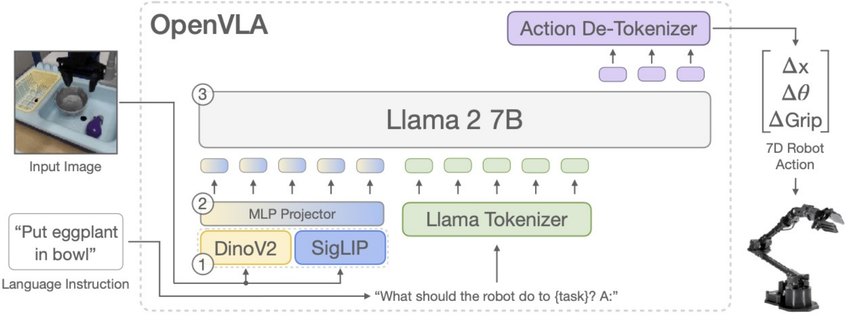
Embodied AI under the era of large models.



香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

An Introduction to Embodied AI

Develop a Vision Language Agent (VLA) to learn generalist policies for robotic control.



Kim, Moo Jin, et al. "OpenVLA: An Open-Source Vision-Language-Action Model." arXiv preprint (2024).

An Introduction to Embodied AI

Embodied AI **in the Pass**:

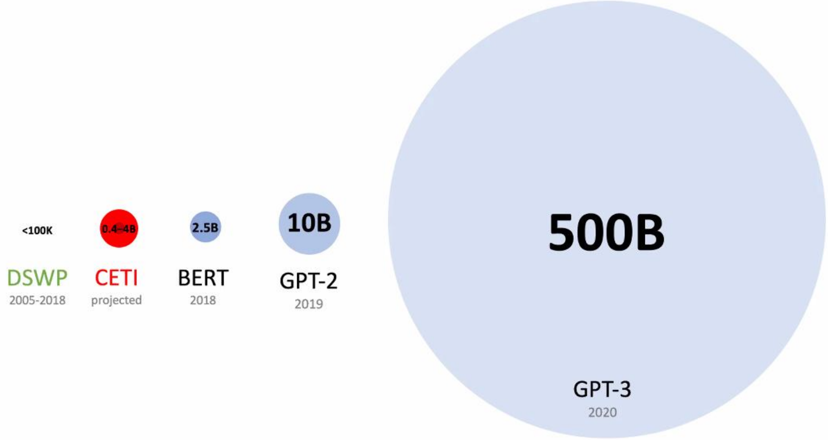
- Goal: Task-Specific Agent.
- Observation: Single Modality.
- Environment: Virtual Environment.
- Methods: Reinforcement Learning, Motion planning, and Optimization.

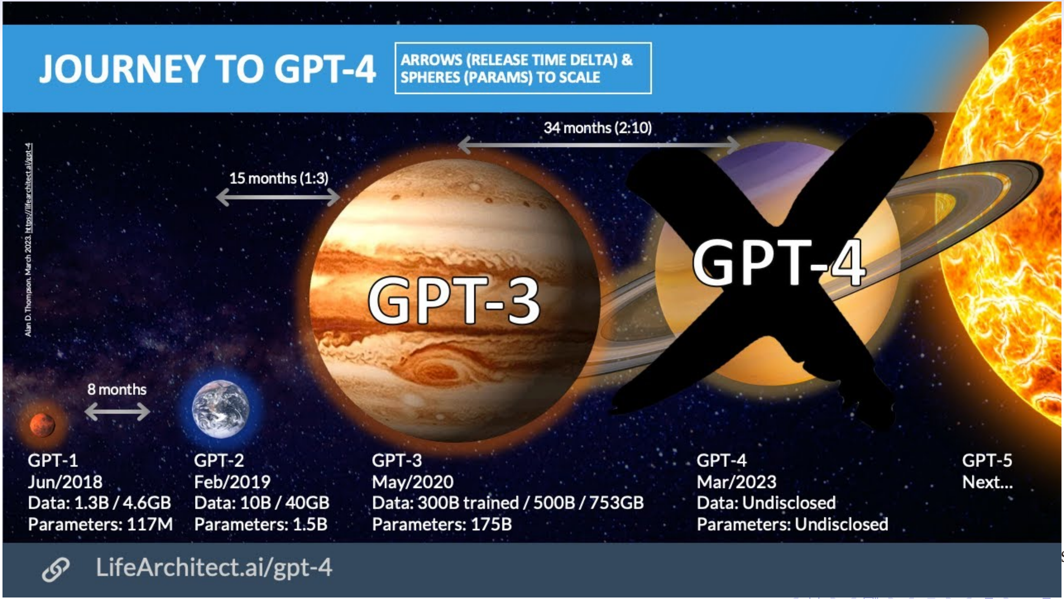
Embodied AI **Nowadays**:

- Goal: Generalist Agent.
- Observation: Multi Modality.
- Environment: Realistic Environment.
- Methods: Reinforcement Learning, Large Multimodal and Decision Model.

Lessons from LLM: Your Data's Size Matters

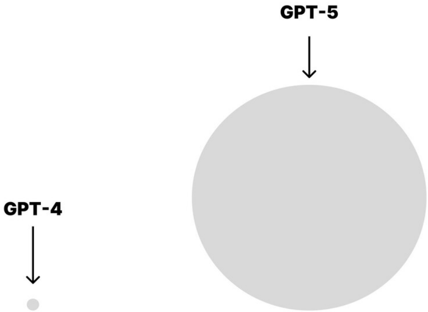
By exhausting more data, one can train strong LLM!




JOURNEY TO GPT-4
ARROWS (RELEASE TIME DELTA) & SPHERES (PARAMS) TO SCALE
Alan D. Thompson. March 2023. https://lifearchitect.ai/gpt-4
34 months (2:10)
15 months (1:3)
8 months
GPT-3
GPT-4
GPT-1
Jun/2018
Data: 1.3B / 4.6GB
Parameters: 117M
GPT-2
Feb/2019
Data: 10B / 40GB
Parameters: 1.5B
GPT-3
May/2020
Data: 300B trained / 500B / 753GB
Parameters: 175B
GPT-4
Mar/2023
Data: Undisclosed
Parameters: Undisclosed
GPT-5
Next...
LifeArchitect.ai/gpt-4

Lessons from LLM: Your Data's Size Matters

Rumors suggest that to update from LLM to VLM, GPT-4 has nearly consumed all the available data. What about "GPT5"?

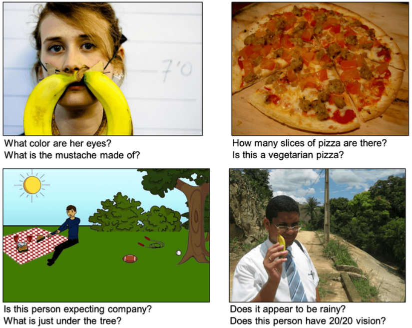


港中文大學（深圳）
Chinese University of Hong Kong, Shenzhen

Data Collection for VLA

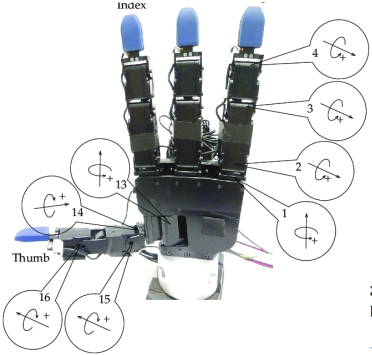
LLM and VLM Training Data:

- Language and image data (e.g., VQA data) that are commonly available.



VLA Training Data:

- Robotic control skills (e.g., 16 DOF Joints) that are less common.



深 圳)
Hong Kong, Shenzhen

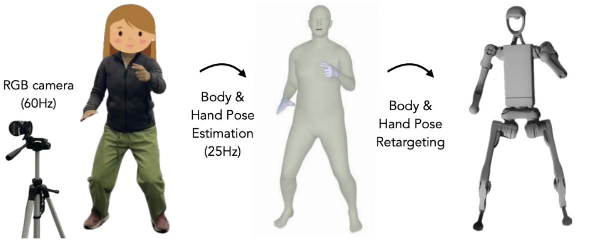
Data Collection for VLA

Manual Tele-Operation:

- Manually control a robot to finish tasks with wearable equipment.

Shadowing and Retargeting

- Use a camera for estimating poses, retarget them to robotic movements.



Data Collection for VLA

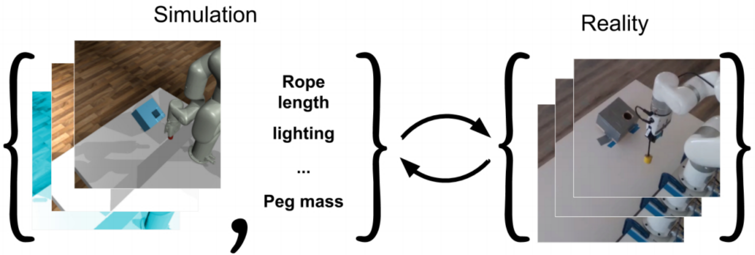
"In this manner, can we generate trillions of data for support VLA training?"

"We made some progress, but not sure if it is tractable and efficient."

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

A Sim-to-Real Approach to Embodied AI

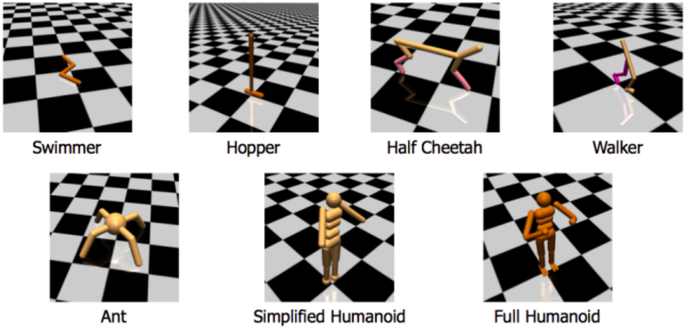
Instead of collecting data from the real world, can we generate data from the simulated environments?



A Sim-to-Real Approach to Embodied AI

Limitations of the current Sim-to-Real.

- Lack of diversity in the operating robots (e.g., MuJoCo).



A Sim-to-Real Approach to Embodied AI

Limitations of the current Sim-to-Real.

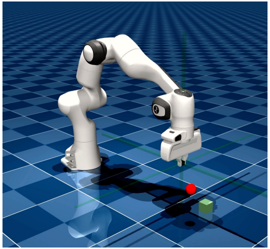
- The number of simulated tasks is limited.

A Sim-to-Real Approach to Embodied AI

Limitations of the current Sim-to-Real.

- The complexity between the simulated and real environment is significant.

Simulation.



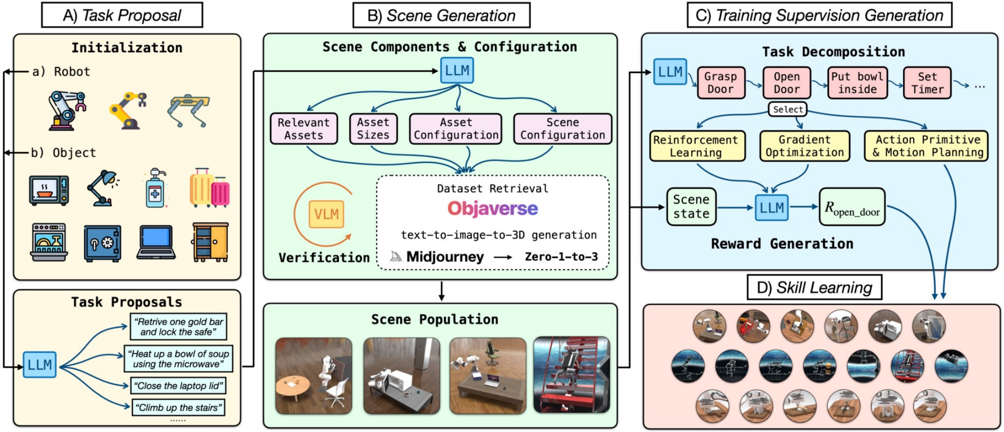
Realistic Office.

Realistic Kitchen.

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

A Sim-to-Real Approach to Embodied AI

Sim-to-Real road-map in Embodied AI (Automated Skill Discovery).

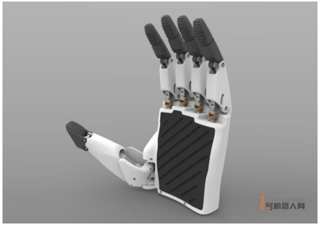


Wang, Yufei, et al. "Robogen: Towards unleashing infinite data for automated robot learning via generative simulation." arXiv preprint

Task Proposal

- Load the robot and its dynamics (e.g., the Degree of Freedom (DoF), size, and visual texture) to the simulator.

Realistic Dexterous Hand



Realistic Robot Arm

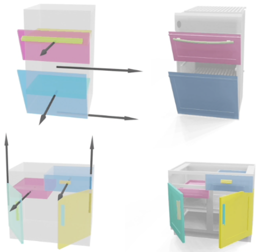
Simulated Arm and Hand

Task Proposal

- Load the 3D objects database to the simulator or generate more complex objects.

Objaverse-XL: An Open Dataset of Over 10 Million 3D Objects.

CAGE: generating 3D articulated objects in a controllable fashion.

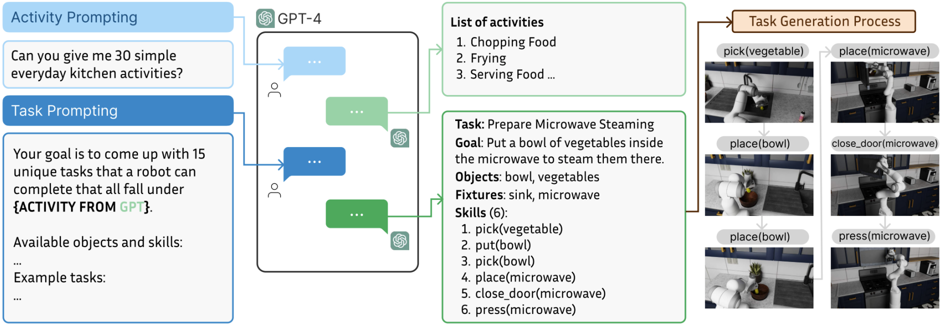


Deitke, Matt, et al. "Objaverse-xl: A universe of 10m+ 3d objects." NeurIPS 2024.
Liu, Jiayi, et al. "CAGE: Controllable Articulation GEneration." CVPR 2024.

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Task Proposal

- Task proposal and decomposition with LLM.



Nasiriany, Soroush, et al. "RoboCasa: Large-Scale Simulation of Everyday Tasks for Generalist Robots." arXiv preprint

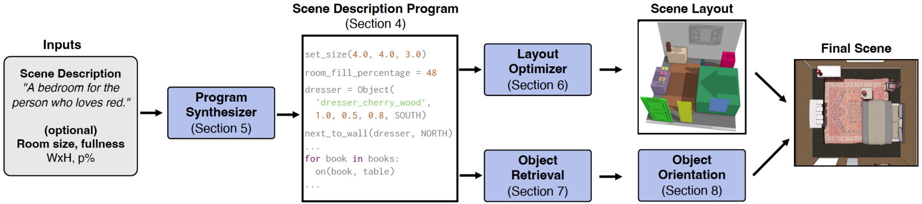
Scene Generation

Generating indoor scenes in response to text prompts:

Aguina-Kang, Rio, et al. "Open-Universe Indoor Scene Generation using LLM Program Synthesis and Uncurated Object Databases." arXiv preprint.

Scene Generation

Generating indoor scenes in response to text prompts:



Aguina-Kang, Rio, et al. "Open-Universe Indoor Scene Generation using LLM Program Synthesis and Uncurated Object Databases."

arXiv preprint.

Scene Generation

Generating realistic and diverse scenes with Robocasa.

Skill Discovery: Training Supervision Generation

Given the proposed task and generated scenarios, it is time to discover useful skills with the Reinforcement Learning (RL) algorithm.

- Skills refer to a policy that solves a specific under a specific scenario. This skill can be embedded in the trajectory $\psi^o = (s_0, a_0, s_1, a_1, \ldots, s_{H_m}, a_{H_m})$ where:
  - State s encloses multi-modal observations, including 3d Cloud, RGB images, language instructions, and tactile as well as force torque signals (or other proprioception signals).
  - Action a refers to specific control signals, e.g., the torques that can be applied to each Degree of Freedom (DoF) in a robot.

Skill Discovery: Training Supervision Generation

The RL objective can be generally represented as:

$$J(\pi) = \mathbb{E}_{\mu_0, P_T, \pi}\left[\sum_{t=0}^{\infty} r(s_t, a_t) + \beta H[\pi(a_t|s_t)]\right] \; s.t. \; D_f(d^{\pi} \| d^{E}) \leq \varepsilon$$

- D_f indicates distributional divergence (KL-divergence, Wasserstein divergence).
- d^E and d^{π} refer to the occupancy measures of the expert and learned distribution.

Solving the problem while aligning with the expert's preference or style.

Embodied AI invites extra challenges!!!

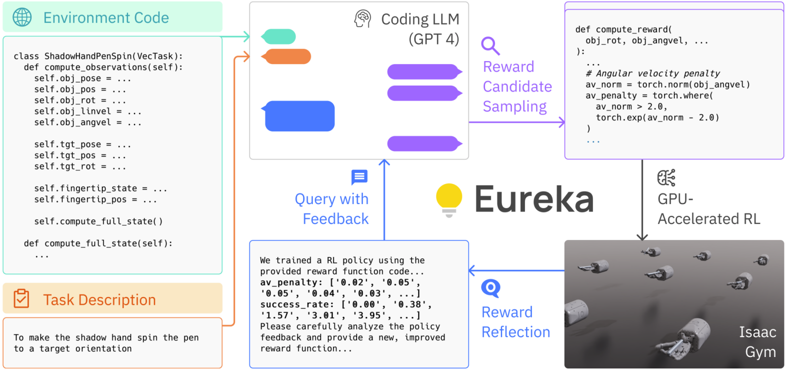
Challenge 1: Designing the Reward Function

The reward function $r(s_t, a_t)$ remained undefined in many embodied tasks.

- Naive rewards: "rewards = is_success".
  - Significant sparsity: Requires extensive exploration and makes learning from sparse rewards challenging.
- Manually rewards: manually design rewards for every tasks.
  - Tractability issues: Relies excessively on human involvement, diminishing the efficiency of learning across a substantial number of tasks.

Challenge 1: Designing the Reward Function

- Automated Rewards Design: relying on AI feedback from LLMs (e.g., eureka).



Ma, Yecheng Jason, et al. "Eureka: Human-Level Reward Design via Coding Large Language Models." ICRL 2024.

Challenge 1: Designing the Reward Function

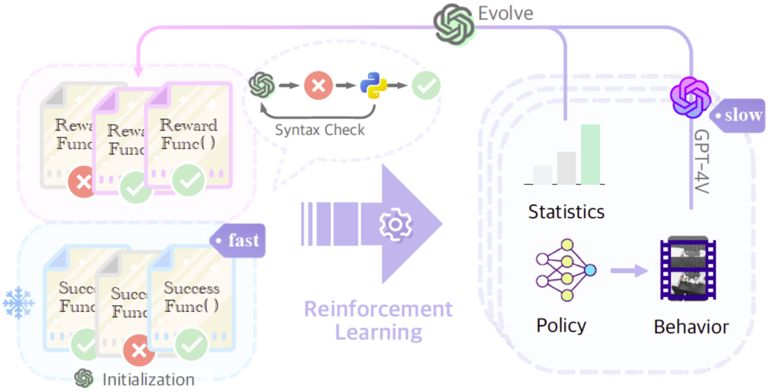
- Automated Rewards Design: current LLM solver has several significant limitations.
  - Complex environments. Task contexts include multiple objects, complicated layouts, and various relations among objects.
  - Multi-modal inputs. Robotic observations include 3d Cloud, RGB images, tactile, and force-torque signals (or other proprioception signals).
  - LLM issues, Standard LLMs may lack proficiency in designing reward models for robotic tasks.

Challenge 1: Designing the Reward Function

Handling Multi-Modal Contexts with Hierarchical Reward Design:

- Utilize low-speed VLM (e.g., GPT 4o) to comprehend the context and initialize the reward function.
- Utilize high-speed LLM (e.g., LLM) to refine the reward functions via evolutionary computation.
- Iterative update the reward function until solving the task at high efficiency (In-context learning).

Challenge 1: Designing the Reward Function



Zhao, Xufeng, Cornelius Weber, and Stefan Wermter. "Agentic Skill Discovery." arXiv preprint arXiv:2405.15019 (2024).

Challenge 1: Designing the Reward Function

Fine-tuning LLM for for more reliable Reward Design:

- The embodied AI environment can label the reward function with feedback (success, speed, and safety).
- Fine-tuning an open-source LLM (Llama 3) for designing the reward functions with the feedback via RLHF.

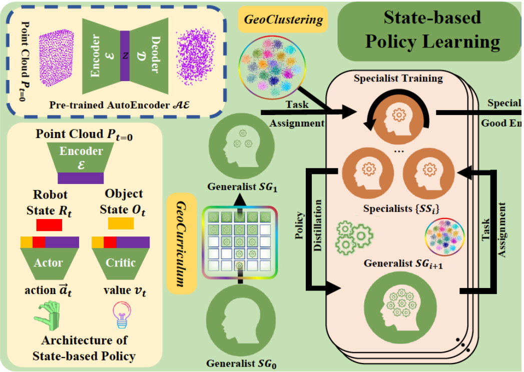
Challenge 2: Scaling to Multiple Tasks

Scaling the RL solver to multiple tasks (typically thousands of tasks) is challenging.

- Maintain an independent RL solver for each task.
  - Computational Intractable. Consuming too much time and computing power.
- Train a RL Solver (Meta RL) for all the tasks:
  - Significant Diversity. A task typically involves different skills and objects, which are difficult to master with only one agent.

Challenge 2: Scaling to Multiple Tasks

- Iterate between generalist and specialist policies.



Wan, Weikang, et al. "Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning." CVPR 2023.

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Challenge 2: Scaling to Multiple Tasks

The current iterative generalist-specialist learning has lots of space for improvement

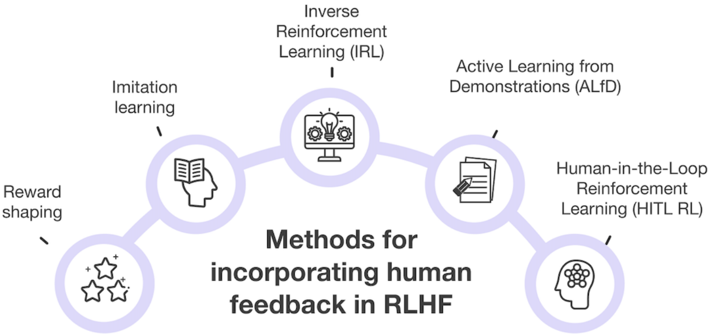
- Sharing Which Information: What types of information can be shared among agents? Is it possible to create an information bottleneck to control the sharing of representations?
- Sharing With Which Agents: How can we determine which agents should share information? Can we develop an efficient mechanism to identify the appropriate targets for information sharing?

Challenge 3: Aligning to Expert Preference

The learned skills must be consistent with human preference.

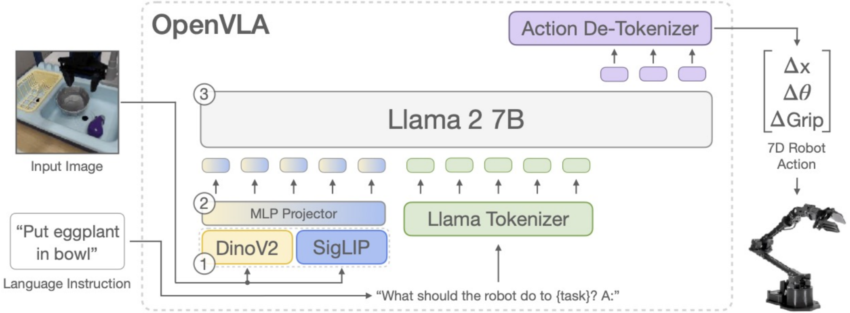
香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Challenge 3: Aligning to Expert Preference



Skill Distillation into a VLA

Distill the skills into a Vision Language Agent (VLA) to learn generalist policies for robotic control.



Kim, Moo Jin, et al. "OpenVLA: An Open-Source Vision-Language-Action Model." arXiv preprint (2024).

Skill Distillation into a VLA

Discrete Action Tokenization:A robotic arm's 7-DOF continuous control vector $[0.25, 0.1, -0.05, \ldots]$ might be quantized $\rightarrow [token_{32}, token_{581}, token_{9}, \ldots]$.

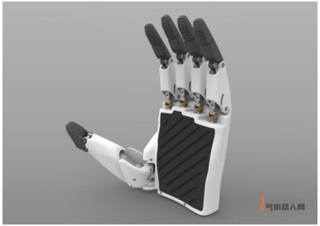
- Actions (which might originally be continuous, e.g., robot joint positions or velocities) are quantized into a finite vocabulary of discrete tokens.
- Each token corresponds to a meaningful or learned primitive action.
- The model then predicts sequences of action tokens just as language models predict sequences of words.
- After generation, these tokens are decoded back into executable motor actions.

Skill Distillation into a VLA

Phase 1: Manipulation Tasks.

- Robot Types : Dexterous hands and robot arms.
- Task: Sim2Real deployment, adapting RL policy to multi-objects manipulation.
- Platform: DexSim simulator and real robots.

Inspire Dexterous Hand



Rokae Robot Arm

DexSim Simulator

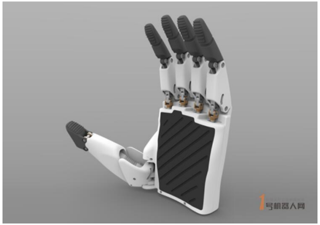
ng, Shenzhen

Skill Distillation into a VLA

Phase 2: Mobile Manipulation Tasks

- Robot Types: Humanoid robot.
- Task: Sim2Real deployment, adapting RL policy to locomotion and manipulation.
- Platform: DexSim simulator and real robots.

Inspire Dexterous Hand



Unitree H1

HumanoidBench Simulator

SimpleVLA-RL

SimpleVLA-RL: A simple yet scalable RL framework for VLA models.

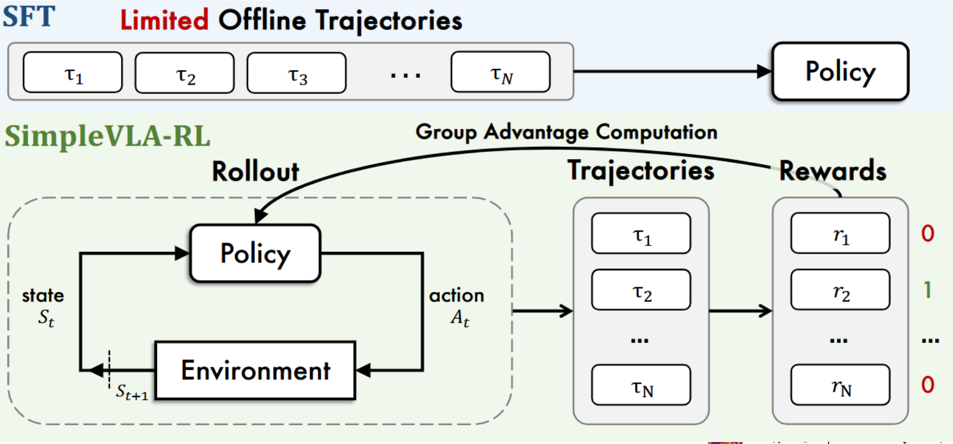
- Key idea: Reuse LLM-style RL (e.g., GRPO) for VLA by
  - Discretizing actions into action tokens.
  - Running closed-loop rollouts in physics simulators.
- Goal: Improve a pretrained VLA policy using online RL in simulation.

From "RL for reasoning tokens" to "RL for action tokens".

SimpleVLA-RL

- Backbone: OpenVLA-style model (vision encoder + LLM) with an action head.
- Two-stage training:
  - Stage 1: Supervised fine-tuning (SFT) on robot demonstrations.
  - Stage 2: Online RL with binary outcome reward (success / failure).

SimpleVLA-RL



MDP Formulation for VLA

SimpleVLA-RL uses a standard MDP for embodied tasks:

- State s_t:
  - RGB images from one or more cameras.
  - Proprioception (joint angles, end-effector pose, gripper state).
  - Language instruction g (e.g., "put the blue bowl on the shelf").
- Action a_t:
  - Low-level control is discretized into action tokens.
  - VLA outputs a distribution $\pi_\theta(a_t \mid s_t, g)$ over these tokens.
- Transition $P(s_{t+1} \mid s_t, a_t)$:
  - Simulated physics (e.g., Isaac / RoboSuite-like robot simulators).
- Reward r_t:
  - Outcome-only: $r_t = 0$ for all $t < T$, and
  - $r_T = 1$ if task succeeds, 0 otherwise.

RL Formulation for VLAs: Objective

Trajectory:

$$\tau = (s_0, a_0, s_1, a_1, \ldots, s_H)$$

Outcome-based reward:

$$R(\tau) \in \{0, 1\}$$

Success/failure of the entire long-horizon task.

Optimization objective:

$$\max_{\theta} \ \mathbb{E}_{\tau \sim \pi_\theta}[R(\tau)]$$

subject to:

- No shaped rewards.
- No value function (critic-free).

Policy Update (Group-wise Advantage)

- For each initial state / task prompt, sample a group of G trajectories

$$\{\tau_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}$$

- Outcome reward: $R(\tau_i) \in \{0,1\}$ (success / failure).
- Group-normalized advantage:

$$A_i = \frac{R(\tau_i) - \overline{R}}{Std[R(\tau_1),\dots,R(\tau_G)] + \varepsilon}$$

- Intuition: compare each trajectory to others in the same group
  - Successful trajectories get positive advantages.
  - Failed trajectories get negative advantages.

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Policy Update (GRPO Loss)

Token-level GRPO loss over all action tokens $a_{i,t}$ in trajectory τ_i:

$$L_{GRPO} = -\mathbb{E}\left[clip\left(\frac{\pi_\theta(a_{i,t} \mid s_{i,t})}{\pi_{\theta_{\text{old}}}(a_{i,t} \mid s_{i,t})}, 1-\varepsilon_{\text{low}},\ 1+\varepsilon_{\text{high}}\right) A_i\right]$$

Properties:

- Critic-free: no value function is learned.
- No explicit KL penalty to a reference policy:
  - simpler implementation, lower memory usage;
  - allows larger policy updates when advantages are large.
- Asymmetric clipping ($\varepsilon_{\text{high}} > \varepsilon_{\text{low}}$)
  - more aggressive updates for good actions (*ClipHigher*).

Exploration: Dynamic Sampling

Outcome-only reward is sparse $\Rightarrow$ many rollouts fail.

Dynamic Sampling: keep only trajectory groups with mixed outcomes.

- For each prompt, sample G trajectories.
- Discard groups where all $R(\tau_i) = 0$ or all $R(\tau_i) = 1$.
- Ensures:
  - non-zero reward variance,
  - stable group-normalized advantages,
  - useful learning signal.

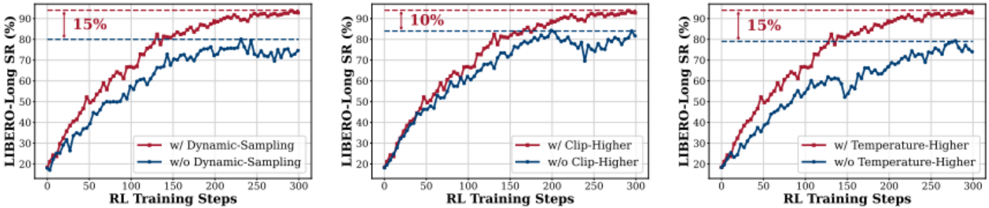
Exploration: ClipHigher & High-Temperature Sampling

Two additional mechanisms to improve exploration:

1. Asymmetric clipping ("ClipHigher")
   - PPO-style ratio clipping:
   $$r \in [1 - \varepsilon_{\text{low}},\ 1 + \varepsilon_{\text{high}}]$$
   - Use $\varepsilon_{\text{high}} > \varepsilon_{\text{low}}$ to allow stronger updates for good trajectories.
2. Higher sampling temperature
   - Use higher temperature for action token sampling during rollouts.
   - Produces more diverse behaviors.
   - Dynamic Sampling filters useful ones.

Exploration: Significant improvement

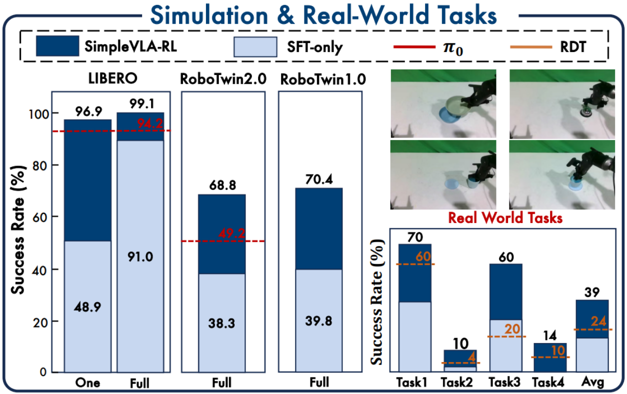
Each mechanism contributes to VLA Performance:



(a) Dynamic Sampling **(b)** Clip Higher **(c)** Higher Rollout Temperature

Empirical Results: LIBERO & RoboTwin

SimpleVLA-RL consistently improves VLA performance.



港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Generalization: Data Scarcity & Sim2Real

SimpleVLA-RL enables strong generalization:

One-trajectory SFT

- Only one demonstration per task.
- RL in simulation recovers most of the full-data performance.

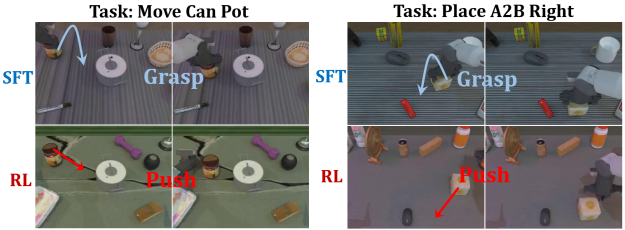
Sim-to-Real transfer

- Policies trained entirely in simulation.
- Higher success rate on real robot compared to SFT.

Generalization: Data Scarcity & Sim2Real

SimpleVLA-RL enables strong generalization: Emergent strategy: "Pushcut"

- RL discovers pushing-based solutions not present in demos.
- Shows ability to learn new behaviors beyond imitation.



香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

Question and Answering (Q&A)