CurricuLLM: Automatic Task Curricula Design for Learning Complex Robot Skills using Large Language Models

Kanghyun Ryu, Qiayuan Liao, Zhongyu Li, Koushil Sreenath, Negar Mehr

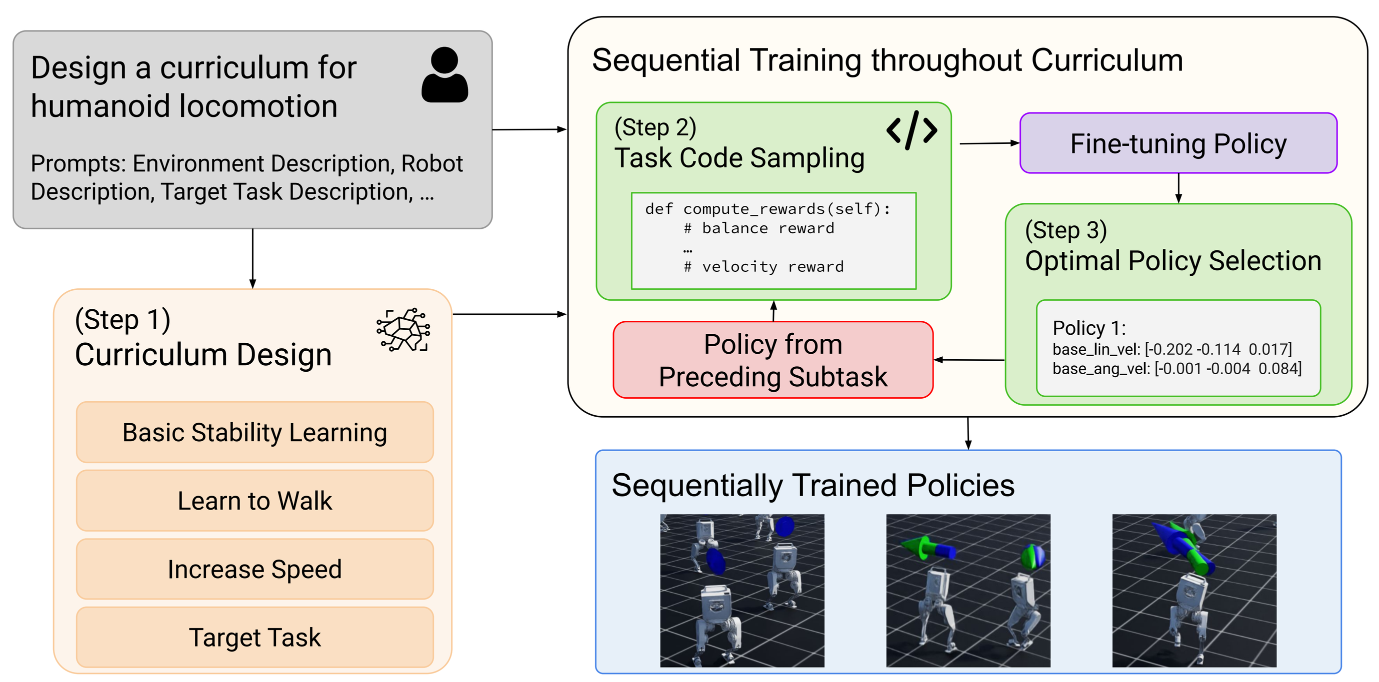
Berkeley UNIVERSITY OF CALIFORNIA · ICON · HYBRID ROBOTICS

Introduction

We introduce CurricuLLM, which leverages the reasoning and coding capabilities of LLMs to design task curricula for complex robotic skills.

Key insights:

- LLM's task decomposition skills can be used for autonomously generating sequences of subtasks that helps learning complex target task.
- LLM can design effective reward functions and goal distributions for each subtask using its code writing ability.



Backgrounds

Classic Task Curriculum Design

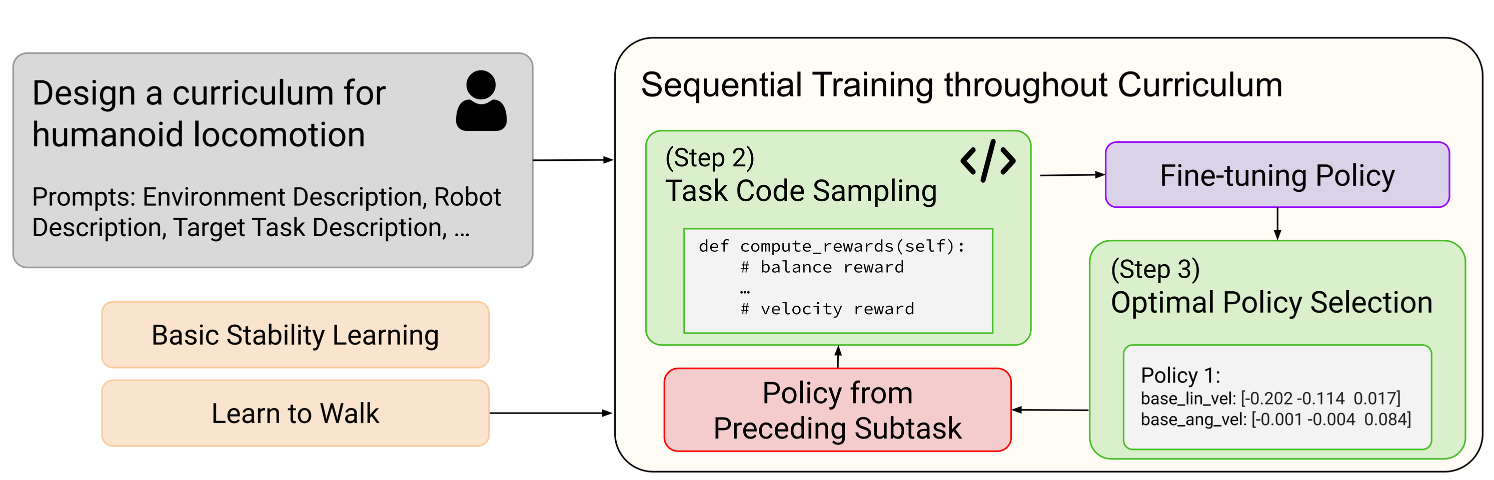
- ✘ Manually designed by human experts, Domain specific
- ✘ Restricted to predefined, limited set of tasks

LLMs in Robotics

- ✔ Automatic task planning and skill decomposition using general world knowledge
- ✔ Translation of natural language to reward function or environment code using programming skills

Our approach: *CurricuLLM*

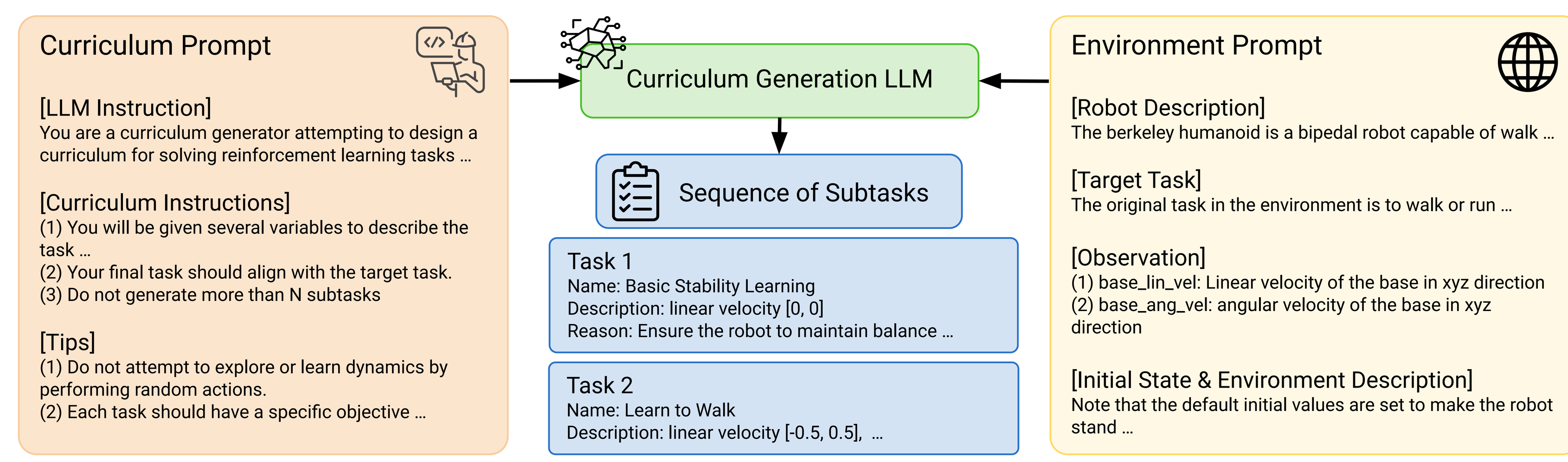
- Step 1: Generate a sequence of subtasks in natural language form to aid target task learning.
- Step 2: Translate natural language description of each subtask in executable task code, which include the reward function code and goal distribution code.
- Step 3: Evaluate trained policies based on trajectory rollout and subtask description.



Detailed Algorithm

Step 1: Curriculum Design

- A curriculum generation LLM receives natural language descriptions of the robot, environment, and target task to generate sequences of language descriptions for the task sequence curriculum.

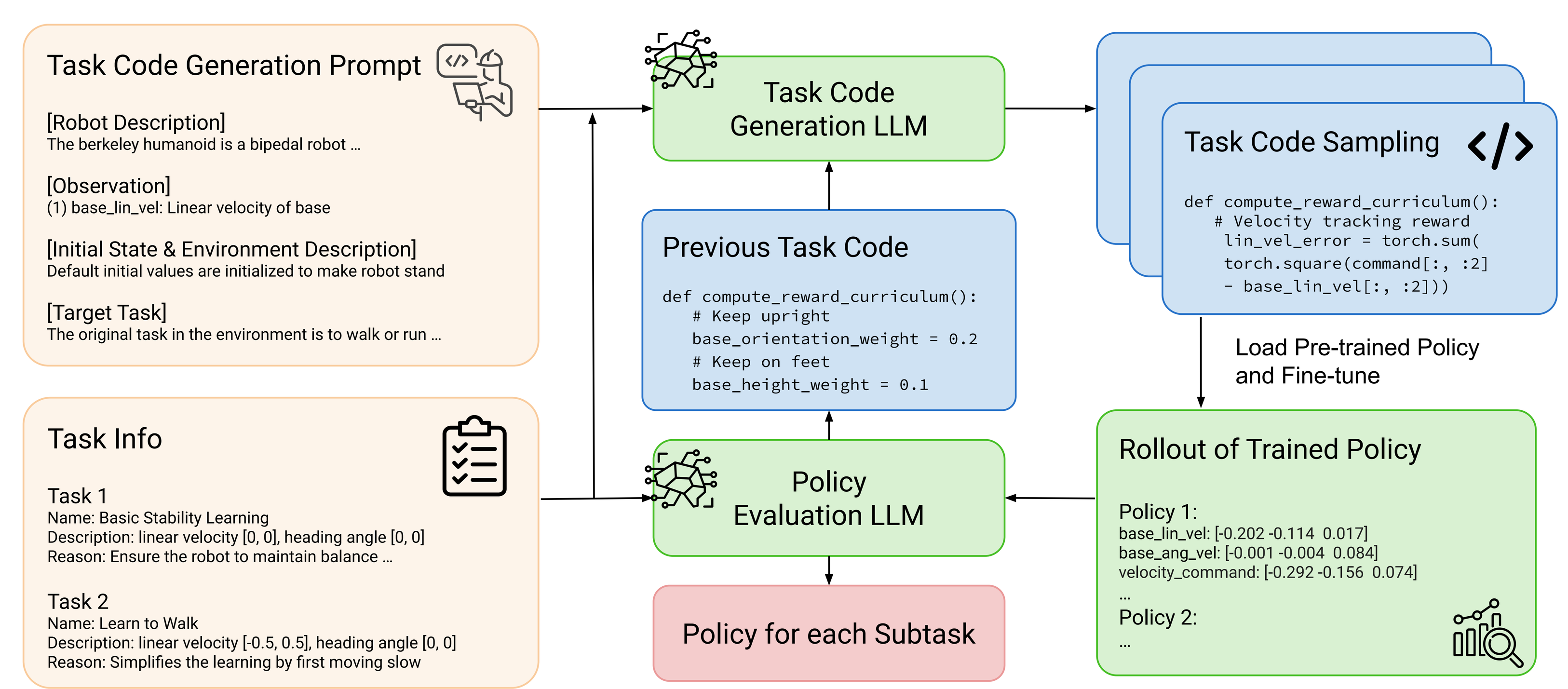


Step 2: Task Code Sampling

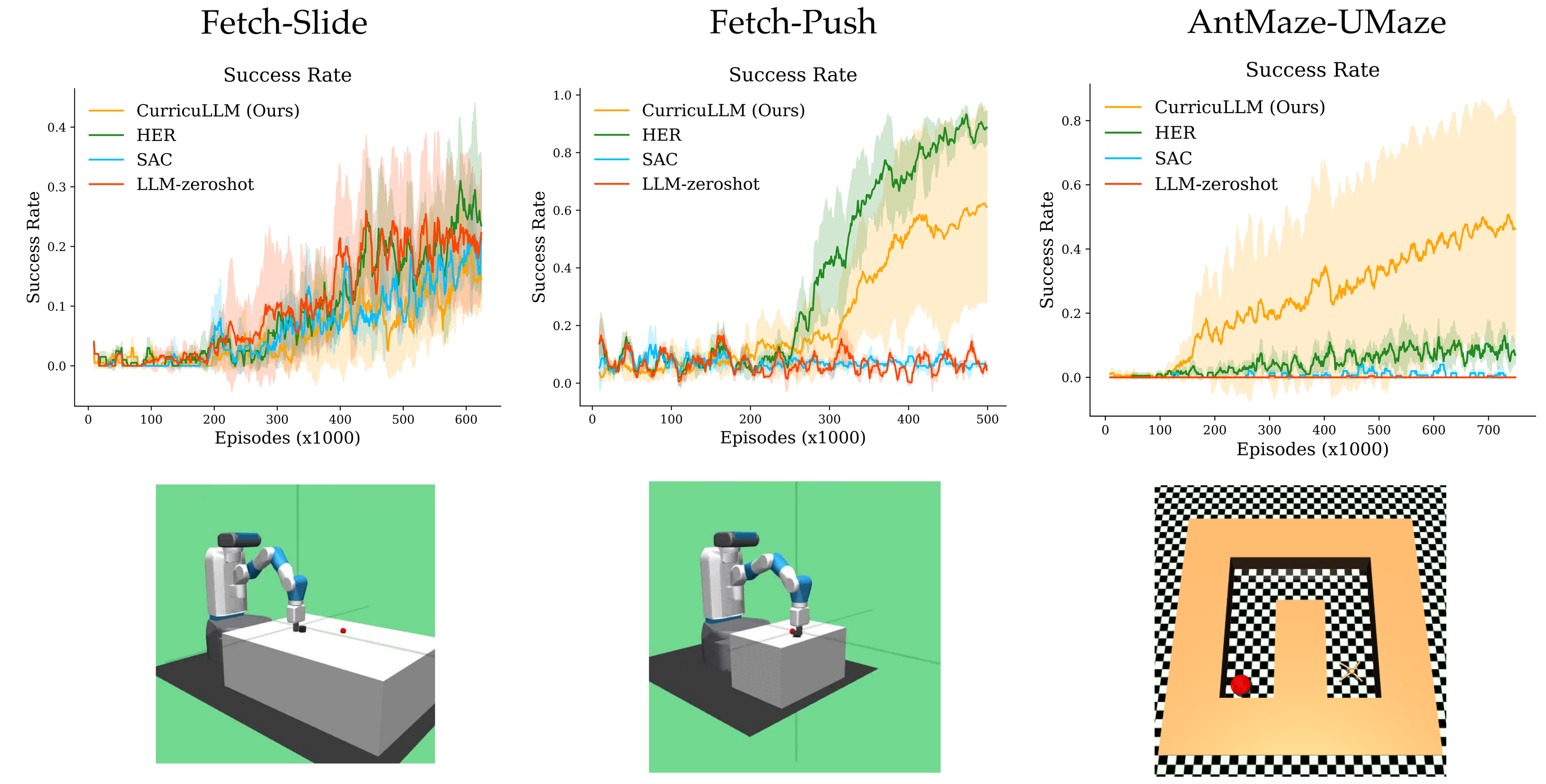
- A task code generation LLM generates samples of task code candidates for the given subtask description. These are used to fine-tune the policy trained for the previous subtask.

Step 3: Optimal Policy Selection

- An evaluation LLM evaluates policies trained with different task code candidates to identify the policy that best aligns with the current subtask. Selected policy is used as a pretrained policy for the next subtask.



Simulation Results



CurricuLLM generates higher Quality Reward Function

LLM-zeroshot generated reward for AntMaze

```
def compute_reward_curriculum(self):
    # Define reward for reaching the goal
    success_reward_weight = 10.0
    success_reward = 0.0
    if goal_distance < 0.45:
        success_reward = 1.0

    # Calculate total reward
    reward = success_reward_weight * success_reward
```

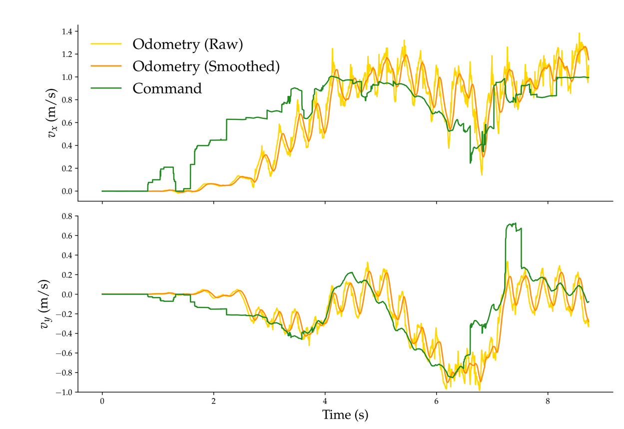
CurricuLLM generated reward for AntMaze

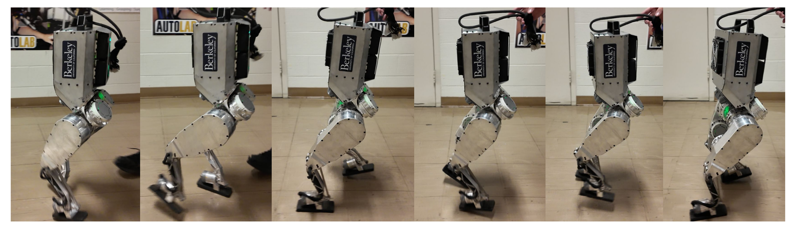
```
def compute_reward_curriculum(self):
    # Calculate the magnitudes
    velocity_magnitude = np.linalg.norm(torso_velocity)
    angular_velocity_magnitude =
        np.linalg.norm(torso_angular_velocity)
    # As goal_distance is received as an array but
    expected to be treated as scalar
    goal_distance_magnitude
        = np.linalg.norm(goal_distance)
    # Weighting parameters setup reflecting curriculum
    learning
    velocity_weight = 0.15
    # Substantial reduction to focus on goal
    achievement
    angular_velocity_weight = 0.15
    # Maintain orientation control importance
    goal_distance_weight = 0.5
    # Continuing to incentivize movement towards goal,
    but with lesser intensity due to the new success
    condition
    success_reward_weight = 2.0
    # High emphasis on reaching close proximity to the
    goal
    ...
```

Reward function from CurricuLLM captures diverse behaviors from learned subtasks.

Facilitates the efficient learning of the target task.

Berkeley Humanoid Experiments





Conclusion

- We introduce CurricuLLM, an automated task curriculum generator using LLMs.
- CurricuLLM successfully created task curricula for diverse robotics tasks in manipulation, navigation, and locomotion.
- We validated the policy learned with CurricuLLM in real-world humanoid locomotion task.

References

Full text available on ArXiv at https://arxiv.org/pdf/2409.18382

[1] Y. J. Ma, W. Liang, G. Wang, D.A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, "Eureka: Human-level reward design via coding large language models," in The Twelfth International Conference on Learning Representations, 2024.

[2] W. Liang, S. Wang, H.J. Wang, Y. J. Ma, O. Bastani, and D. Jayaraman, "Environment curriculum generation via large language models," in 8th Annual Conference on Robot Learning, 2024.

[3] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor, and P. Stone, "Curriculum learning for reinforcement learning domains: A framework and survey," Journal of Machine Learning Research, vol. 21, no. 181, pp. 1–50, 2020.

[4] Q. Liao, B. Zhang, X. Huang, X. Huang, Z. Li, and K. Sreenath, "Berkeley humanoid: A research platform for learning-based control," arXiv preprint arXiv:2407.21781, 2024.