Adaptive Action Chunk Selector

Team: Ruopei Chen, Ke Wang, Yazhou Zhang TA: Marcel Villasevil
Course: CS224R Deep Reinforcement Learning

LEFT: bimanual insertion success demo
RIGHT: bimanual insertion failure demo

BACKGROUND

Action Chunking with Transformers (ACT) improves IL by predicting multi-step action sequences, reducing compounding errors and producing smoother motions. On the ALOHA platform, ACT demonstrated strong performance on complex, fine-grained manipulation tasks using only low-cost hardware and vision input. However, ACT assumes a fixed action chunk length k across all timesteps, which may not be ideal for dynamic, contact-rich interactions.

GAP IN PRIOR WORK

- ACT uses a fixed chunk size k, forcing a trade-off:
  - Small k: reactive but noisy and compute-heavy
  - Large k: smooth but less responsive to changes
- This fixed k treats all states as equally predictable, ignoring task context or temporal complexity.
- Real-world demonstrations often involve non-uniform structure (e.g., pauses, multi-step coordination), requiring variable control granularity.
- Without fixed k, the agent struggles to generalize to tasks with mixed dynamics or fine-grained feedback needs.

OBJECTIVE

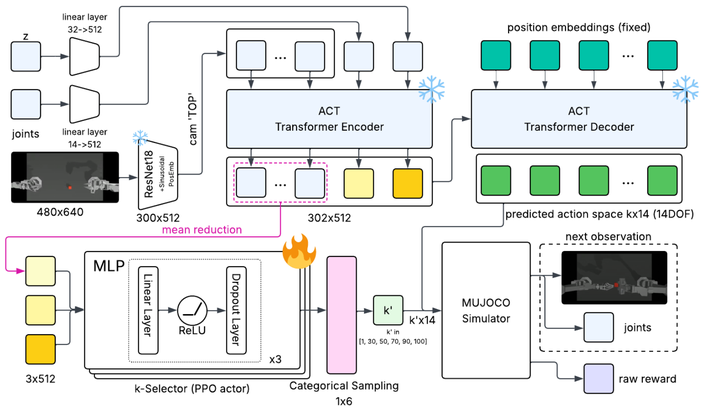
- Enable adaptive control by training a k selector that dynamically adjusts action horizon k based on current state and observations.
- Balance reactivity and smoothness:
  - Use smaller k for fine-grained, contact-rich phases.
  - Use larger k for stable, coarse movement phases.
- Evaluate on ALOHA manipulation tasks to benchmark performance.
- Impact: Improve policy generalization and robustness in real-world robotics, especially in tasks combining fine-grained manipulation with long-horizon planning.

REFERENCES

[1] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware," arXiv:2304.13705, Apr. 2023.

METHOD

- online PPO based k'-selector

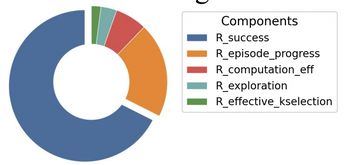


- Raw Reward from MUJOCO Simulator

  $r_{timestep\_i} \in 0, 1, 2, 3.4$ evaluated according to robot and task object contact state

  $$R_{raw\_reward} = \sum_{i=1}^{k'} r_{timestep\_i}$$

- Reward Design for PPO



$$R_{episode\_progress} = \min(\frac{R_{raw\_reward}}{env\_\max\_reward \times 100}, 1.0)$$

$$R_{exploration} = \max(\left\{1 - 2\left|\frac{k'-1}{100} - 0.7\right|\right\}, 0.1)$$

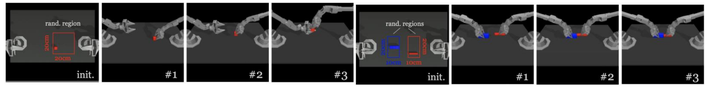
$$R_{effective\_kselection} = R_{episode\_progress} > 0.7 ? \frac{k'-1}{100} : 1\text{-}\frac{k'-1}{100}$$

$$R_{computation\_eff} = \frac{k'}{100}$$

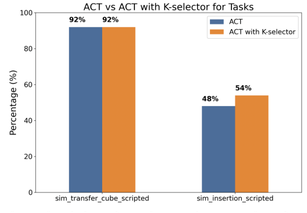
$R_{success}$: Reward signal for completing the task
$R_{episode\_progress}$: Reward for making meaningful progress toward task completion
$R_{computation\_eff}$: Reward for avoiding unnecessary short action chunk
$R_{exploration}$: Reward signal for encouraging diversity in k selection
$R_{effective\_kselection}$: Reward for adaptive k-selection

RESULTS

- Task: LEFT - Transfer cube; RIGHT – Bimanual insertion [1]

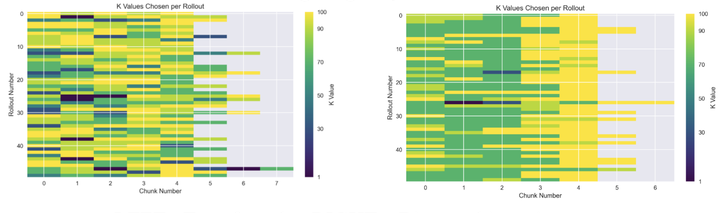


- Task success rate for ACT (k=100) and ACT with K-selector



  - *sim_transfer_cube_scripted*: Both ACT and ACT with K-selector achieve a 92 % success rate. ACT's performance is already near ceiling, so adding the k'-selector yields no further gain.*
  - *sim_insertion_scripted*: Incorporating the k'-selector raises success from 48 % to 54 %, representing a 6 % absolute improvement.*

- Variation in k' selection along single episode for 50 rollouts



LEFT - Transfer cube; RIGHT – Bimanual insertion

Across both tasks, the agent consistently selects smaller action chunk sizes (lower k' values) during phases where the two robot arms must coordinate closely. This behavior is expected, as reducing k′ allows the agent to observe the environment more frequently, increasing safety and precision at critical interaction points.

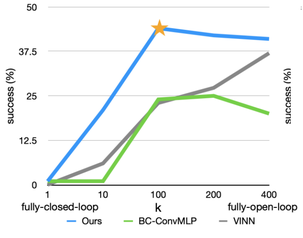
LIMITATION & FUTURE WORK

Dataset

- **Current task** data were generated by **scripts**. The performance difference should be compared further using task data collected by **human**. **
- To accurately evaluate the adaptive chunking mechanism, we need benchmark tasks where the optimal chunk length varies during the task. One proposal is a **multi-phase manipulation task**: for instance, a task that involves a long transport phase (where a long chunk is advantageous to cover distance quickly) followed by a delicate precise phase (where a short chunk or step-by-step control is needed).

ACT Model

- The current ACT model is trained with k = 100. Reference [1] shows that performance rises sharply as k increases from 1 to 100, then tapers off only mildly up to k=400. To maximize flexibility, we propose retraining ACT with k=400, while allowing the k′-selector to choose any chunk length in the range 1≤k′≤400. **



k' selector network improvement

- The ACT was trained to assume all k actions are executed. Introducing k′-selector may cause data in the later stage of the episode out of distribution. One proposal is to train the k′-selector with the ACT decoder end-to-end.

*results were evaluated on 50 rollouts
**results will be shown in the final report