Proposal of SS25 ADLR Project

FAST Tokenization for Time-Discrete Control: Learning with Compressed Action Representations

Ruiming Wu
Technical University of Munich
ruiming.wu@tum.de

Baiheng Li
Technical University of Munich
go76ren@mytum.de

I. Objective

In this project, we want to explore how different ways of representing action sequences affect the learning of control models. We focus on simple simulation environments like **CartPole**, and plan to train two small **Transformer** models using two approaches:

A. a basic **Naïve Binning** method, and

B. the more advanced **FAST tokenization** (which compresses actions using DCT, quantization, and BPE).

Our goal is to compare the two methods in terms of **training speed**, **control performance**, and **sequence prediction accuracy**, and see whether compression-based tokenization like FAST offers clear benefits in this kind of task.

II. Related Work

Many current Vision-Language-Action (VLA) models like π_0 [1] and OpenVLA [2] use simple binning methods to turn continuous robot actions into discrete tokens. This works well for low-frequency tasks, but once the control becomes fast or more fine-grained, these models tend to struggle—mainly because the token sequences get too long and redundant.

To deal with this, FAST (Frequency-space Action Sequence Tokenization) was recently proposed. It compresses action sequences using Discrete Cosine Transform (DCT) and Byte-Pair Encoding (BPE), which helps reduce sequence length and improve training speed. Research showed that combining FAST with models like π_0 or OpenVLA makes them train faster and perform better on more challenging, high-frequency tasks [3]. That's why we're interested in testing this idea in our own setup and comparing it to the more traditional binning approach.

III. Technical Pipeline (estimated)

We will primarily use Python, with OpenAI Gym for simulation and PyTorch for model training. Our technical pipeline is structured as follows:

A. Data Collection

We simulate CartPole using PID controllers with varied parameters to generate state-action trajectories for supervised training.

B. Two Modeling Pipelines

1) tiny π0-FAST:

a) Apply DCT, quantization, and BPE to compress action sequences into discrete tokens.

b) Train a small Transformer (autoregressive) model to predict token sequences from state observations.

c) Decode tokens back into continuous actions for control.

2) tiny π0:

a) Train a small Transformer model to directly predict continuous actions using Flow Matching loss, without tokenization.

C. Evaluation

We compare both models based on:

- Control performance (stability, accuracy)
- Training efficiency (speed, convergence)
- Token compression (length, vocabulary)
- Generalization under small changes in environment dynamics (e.g., mass, damping).

IV. References

[1] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi 0: A vision-language-action flow model for general robot control. arXiv preprint arXiv:2410.24164, 2024.

[2] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. arXiv preprint arXiv:2406.09246, 2024.

[3] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. arXiv preprint arXiv:2501.09747, 2025.

V. Appendix (Technical Flowchart)