2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

# Hierarchical Action Chunking Transformer: Learning Temporal Multimodality from Demonstrations with Fast Imitation Behavior

J. Hyeon Park [1,2,3] Wonhyuk Choi [1,2] Sunpyo Hong [1], Hoseong Seo [1], Joonmo Ahn [1]
Changsu Ha[1], Heungwoo Han [1], and Junghyun Kwon [1]

***Abstract*— Behavioral cloning from human demonstrations has succeeded in programming a robot to generate fine-grained motion, but it is still challenging to learn multimodal trajectories such as with various speeds. This restricts the use of a robot dataset collected by multiusers because the different proficiency of robot operators makes the dataset have diverse distributions of speed. To tackle this issue, we develop Hierarchical Action Chunking Transformer with Vector-quantization (HACT-Vq) to efficiently learn temporal multimodality in addition to fine-grained motion. The proposed hierarchical model consists of a high-level policy to make planning for a latent subgoal and style, and a low-level policy to predict an action chunk conditioned with the latent subgoal and style. The latent subgoal and style are trained as discrete representations so that high-level policy can efficiently learn multimodal distributions of demonstrations and retrieve the mode of fast behavior. In experiments, we set up bimanual robots in both simulation and real-world environments, and collected demonstrations with various speeds. The proposed model with the quantized subgoal and style showed the highest success rates with fast imitation behavior. Our code is available at *https://github.com/SamsungLabs/hierarchical-act*.**

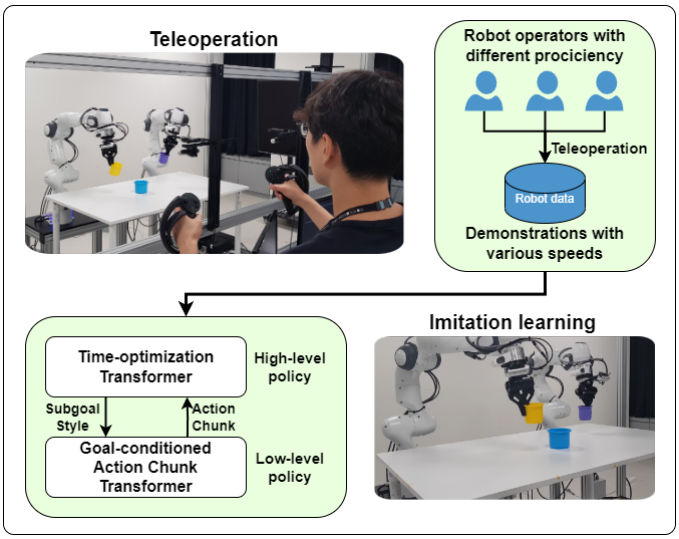


Fig. 1: We joint action data of bimanual robots by teleoperation. As multiusers with different proficiency joined data collection, the dataset has various distributions of speed. We propose a hierarchical behavioral cloning model that can learn the data with various speeds and retrieve fast imitation behavior.

## I. INTRODUCTION

Recently, Behavioral Cloning (BC) with generative models has shown impressive success in imitation learning as the generative models allow to learn multimodal distributions of demonstrations. One of the widely adopted approaches is a transformer decoder with tokenized actions because it can efficiently learn a probabilistic sequential model of demonstrations [1], [2], [3]. Although the transformer shows powerful performance to learn complex multimodality, the tokenized actions make the quality of motions deteriorates.

Instead of tokenizing actions, Action Chunking Transformer (ACT) [4] directly predicts the chunk of continuous actions conditioned with a probabilistic variable where the multimodality can be encoded. This combination of the action chunk and the multimodality encoder showed impressive performance in many complex skills. However, we found that the performance to learn multimodality of ACT was not enough to learn demonstrations with complex multimodality such as various temporal modalities. This made it difficult for us to use a dataset collected by multiusers because various proficiency of teleopreators makes the dataset have various distributions of speed.

To tackle this issue, we propose Hierarchical ACT with Vector-quantization (HACT-Vq) to learn the complex multimodality of demonstrations, especially temporal multimodality. The proposed structure consists of a high-level policy to predict a latent subgoal and style, and a low-level policy to predict continuous actions conditioned with the subgoal and style which are the output of the high-level policy. To efficiently learn multimodal distributions of demonstrations, we train a probabilistic model of the latent subgoal and style as quantized representations via Vector-quantized Variational AutoEncoder (Vq-VAE) [5]. Instead of continuous representations as proposed in ACT, the use of quantized representations for the latent variables improves performances to learn the temporal multimodality of demonstrations. Additionally, we design our model to retrieve fast behavior imitation by predicting the time cost of the sampled latent subgoals and styles, and selecting one of them with the minimum time cost. This selection of imitation mode provides practical convenience for mining expert skills with less time cost from suboptimal demonstrations.

We set up bimanual robots in both simulation and real-world environments. In simulation environments, we collected data via scripted policies where the speeds of demonstrations are intentionally modulated to collect various types

[1] Samsung Research, Samsung Electronics Co. Ltd, Seoul, Korea.
[2] These two authors are contributed equally to this work.
[3] Corresponding author, jh_raph.park@samsung.com

of speed of demonstrations. We show that the proposed HACT-Vq can learn the various types of temporal multimodality with improved success rates and imitation time. In real-world experiments, we collected data via teleoperation where multiusers with different proficiency joined to operate. Likewise, the proposed HACT-Vq shows improved success rates even though the multimodality of the datasets.

## II. Related works

For BC of unstructured multimodal demonstrations, generative models are useful in that they do not assume the tractable posterior distribution of the data. There are many studies on multimodal BC with the generative models such as Variational Autoencoder (VAE), Generative Adversarial Network (GAN), and diffusion model. The BC with VAE [6], [4] can be stably trained and work well with the small number of data, but their generative performance is too weak to learn the complex modality of actions. The BC with GAN [7], [8] can learn complex multimodality, but they use adversarial training which becomes easily unstable with the small number of data. Therefore, they are not available for many real-world scenarios, especially with the high-dimensional system. Diffusion model [9], [10] can be stably trained to generate high-quality actions, but the denoising process used in the model is computationally inefficient in both training and inference, and it is hard to select the intended mode inside the diffusion model.

On the other hand, some works introduce a transformer model [11] as a sequential generative model for BC. As the transformer models needs the discrete tokens for its outputs, the output actions need to be tokenized as quantized values. A simple method for the tokenization is to discretize actions by binning [1], [12], but they can generate jittering motion because of discontinuity. Some works encode continuous actions into discrete representations such as a k-means encoder [2], [13], Vq-VAE [14], and Vq-GAN [3]. The use of discrete representations of actions allows learning complex multimodality of demonstrations, but it degrades the quality of actions, which results fine manipulation hard.

In this research, we propose a transformer model combined with Vq-VAE to learn both multimodality and high-quality actions of demonstrations. The VAE alone has weak generative performance in multimodal actions, and a transformer alone cannot learn continuous high-quality actions. To deal with both multimodality and continuous actions, we train a transformer equipped with Vq-VAE to generate the discrete representations of demonstrations, then train ACT [4] conditioned with the discrete representations to predict continuous high-quality actions.

The original ACT [4] based on vanilla VAE to learn a latent style, but we found that the VAE has weak performance on multimodal demonstrations, especially with various temporal modalities. On the other hand, our use of Vq-VAE increases the multimodal performance of the action model. To the end, our model can retrive multimodal and continuous high-quality actions at the same time. Additionally, the Vq-VAE in the proposed model allows to select the mode for retrieved actions, as it has the finite quantized style. Comparing to the other generative models such as VAE, GAN and diffusion model that use continuous latent probabilistic variable, it is hard to distinguish mode in the latent space. On the other hand, our Vq-VAE encodes the mode of demonstration into the quantized value, so that we can select the mode of fast behavior.

## III. Preliminaries

### A. Behavioral Cloning

Behavioral cloning (BC) is one of the simplest forms of imitation learning. BC mimics expert skills by supervised learning of expert demonstrations. Let us say that a dataset consists of trajectories $\mathcal{D} = \{\tau_1, \tau_2, ...\}$ where the trajectory with length $T$ is a sequence of states and actions $\tau = (s_1, a_1, ..., s_T, a_T)$. The objective function of BC in supervised learning is the probabilistic distance of the action distributions between the dataset $p(a_t|s_t)$ and a training policy $\pi(a_t|s_t)$.

$$\min_{\pi} \mathbb{E}_{(s_t,a_t)\in\mathcal{D}}[D\left(p(a_t|s_t) \parallel \pi(a_t|s_t)\right)] \tag{1}$$

Minimizing this loss makes the policy reproduce expert behavior with the same distribution as in the dataset.

### B. Vector-quantized Variational Autoencoder

Vector-quantized Variational Autoencoder (Vq-VAE) [14] is a variant of VAE that learns discrete representations via vector quantization. Let us have a state encoder $\phi(z|x)$ where $x$ is an input and $z$ is continuous embeddings, and a codebook $(z_1^e, z_2^e, ..., z_n^e)$ where $z_i^e$ is a trainable vector for $i = 1, ..., n$. The continuous embeddings $z$ are mapped into quantized embeddings $z^q$ as the closest vector in the codebook by measuring Euclidean distance.

$$z^q = z_{j^*}^e \text{ where } j^* = \underset{j=1,2,...,n}{\arg\min} ||z - z_j^e||_2^2 \tag{2}$$

The objective of Vq-VAE is to train the encoder and the codebook to provide relevant $z^q$ for a downstream task. The loss function of Vq-VAE is as follows:

$$\mathcal{L}_{Vq} = ||sg[z] - z^q||_2^2 + ||z - sg[z^q]||_2^2 \tag{3}$$

where $sg[\cdot]$ is an operator of stop gradient. This loss is added to the loss of a downstream task so that the gradient from the downstream task can pass to the encoder and the codebook.

## IV. Method

We propose a two-level policy where a high-level policy generates a latent subgoal and style, and a low-level policy predicts a continuous action chunk conditioned with the subgoal and style generated by the high-level policy. We newly propose Time-Optimization Transformer (TO-Transformer) as a generative model for the high-level policy to learn the multimodality of demonstrations and retrieve fast imitation behavior. For the low-level policy, Goal-Conditioned ACT (GC-ACT) is proposed as a similar structure to ACT [4], but we modify ACT to use additional inputs for the latent

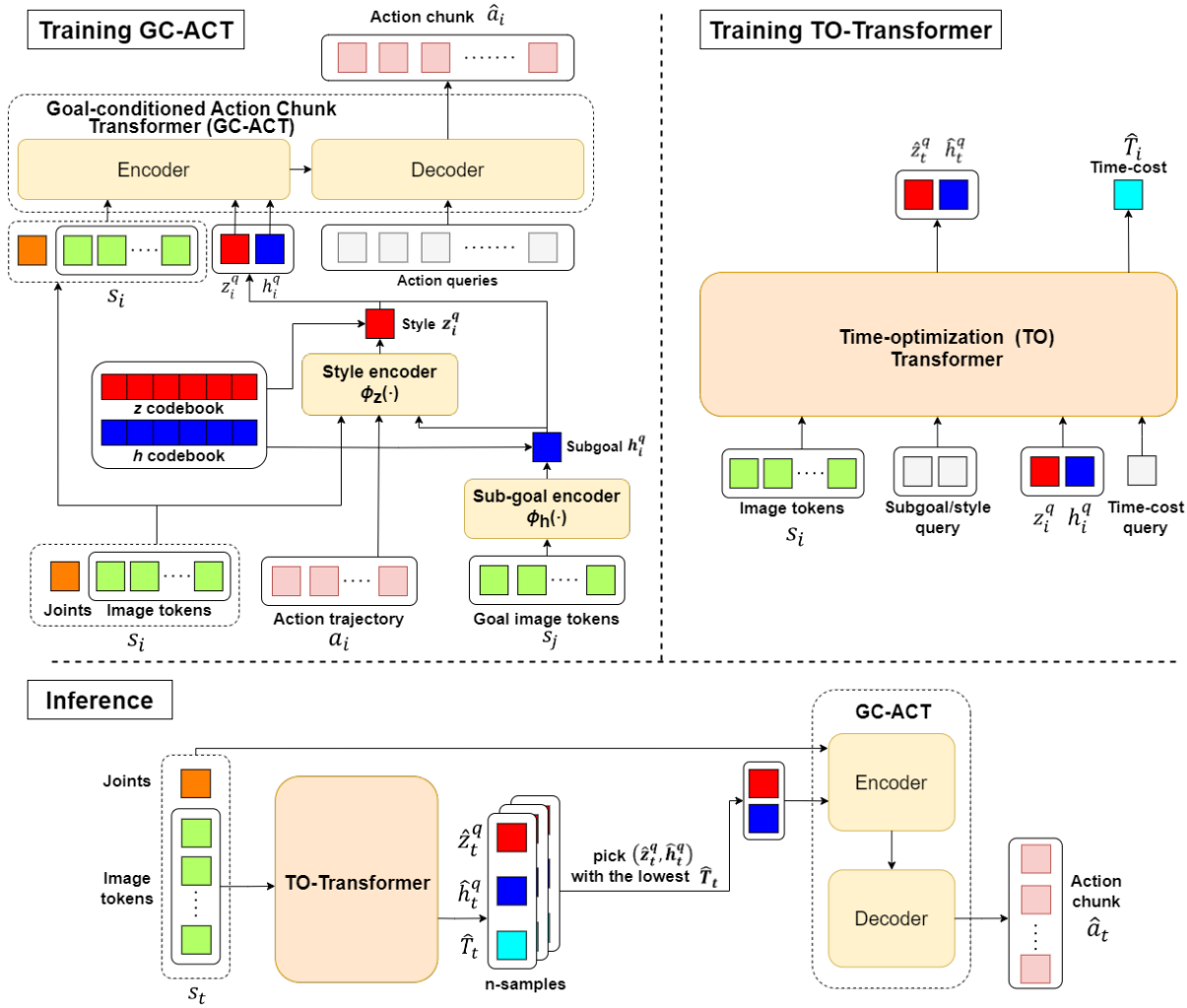


Fig. 2: The overview of the proposed model. GC-ACT is a low-level policy to predict an action chunk conditioned with a latent subgoal and style, and TO-Transformer is a high-level policy to generate the latent subgoal and style. The latent subgoal and style are trained with GC-ACT as discrete representations by codebooks of Vq-VAE.

subgoal and style. Additionally, we use discrete representations for the latent subgoal and style embedded by Vq-VAE [5]. The discrete representations allow the high-level policy to efficiently learn the multimodality of demonstrations. The overview of our entire model is shown in Fig. 2.

### A. *Goal-Conditioned Action Chunking Transformer*

We design GC-ACT $\pi(a_{t:t+k}|s_t; h_t^q, z_t^q)$ as the low-level policy where $s_t$ is a current state, $h_t^q$ is a latent subgoal, $z_t^q$ is a latent style, and $a_{t:t+k}$ is a chunk of $k$ actions. The GC-ACT is a goal-conditioned policy that predicts $a_{t:t+k}$ from the current state $s_t$ and the given goal state $s_g$. The goal state $s_g$ is encoded into the quantized latent variable $h_t^q$ by the auxiliary encoder $\phi_h(\cdot)$, and the $h$ codebook $(h_1^e, h_2^e, ..., h_n^e)$ where $h_i^e$ for $i = 1, ..., n$ is trainable vectors. Furthermore, the policy is conditioned with the latent style vector $z_t^q$ which represents the multimodal styles of the given demonstration. The quantized latent style $z_t^q$ is obtained by encoding the state $s_t$ and the action chunk $a_{t:t+k}$ using the auxiliary encoder $\phi_z(\cdot)$, and the $z$ codebook $(z_1^e, z_2^e, ..., z_m^e)$ where $z_i^e$ for $i = 1, ..., m$ is trainable vectors. The encoders and the codebooks are jointly trained with GC-ACT as a manner of Vq-VAE in training time, but not used in the inference time as $(h_t^q, z_t^q)$ is provided by the high-level policy. Similar to ACT, the structure of the proposed GC-ACT is an encoder-decoder transformer to predict an action chunk $a_{t:t+k}$ instead of a single-step action $a_t$. This use of the action chunk can reduce the compounding errors of BC.

To train GC-ACT, let us sample a tuple of $(s_i, \mathbf{a}_i, s_j)$ from an episode in a dataset where $s_i$ and $s_j$ are states sampled from a trajectory with $i < j$ and $\mathbf{a}_i = a_{i:j}$ is a chunk of actions between $s_i$ and $s_j$. The states consist of images and robot joint positions, and the action is the reference joint positions. For inputs of the transformer, we tokenize input images by flattening the feature map of Resnet-18 [15]. We use $s_i$ as the mixed notation for the images and the images tokens in the later of this paper for explanation convenience. Then, we obtain the continuous embeddings for subgoal $h_i =$

$\phi_h(s_j)$ with the subgoal encoder $\phi_h(\cdot)$. From the continuous embeddings $h_i$, we select the quantized embeddings $h_i^q$ by finding the closest vector in the $h$ codebook as we explained in Eqn (2). To train the mapping between the encoder $\phi_h(\cdot)$ and the $h$ codebook, we use the following loss as we explained in Eqn (3).

$$\mathcal{L}_h = ||sg[h_i] - h_i^q||_2^2 + ||h_i - sg[h_i^q]||_2^2 \tag{4}$$

Likewise, we obtain the continuous embeddings for the latent style $z_i = \phi_z(s_i, \mathbf{a}_i, h_i^q)$ with the style encoder $\phi_z(\cdot)$ which determines the style of actions $\mathbf{a}_i$ to reach the latent subgoal $h_i^q$ from the current state $s_i$. From the continuous embeddings $z_i$, we select the quantized embeddings $z_i^q$ by finding the closest vector in the $z$ codebook. The training loss for the encoder $\phi_z(\cdot)$ and the $z$ codebook is as follows.

$$\mathcal{L}_z = ||sg[z_i] - z_i^q||_2^2 + ||z_i - sg[z_i^q]||_2^2 \tag{5}$$

The use of Vq-VAE instead of VAE provides several benefits. At first, the multimodality of demonstrations is effectively captured in discrete representations compared to VAE. We will further discuss it in our experimental section. Also, discrete representations allow high-level policy to learn a probabilistic sequential model of $(h^q, z^q)$.

Finally, GC-ACT $\pi(\mathbf{a}_i|s_i, h_i^q, z_i^q)$ predicts the continuous action chunk with the encoder-decoder transformer. These inputs are encoded by the transformer encoder, and the transformer decoder predicts the action chunk $\hat{\mathbf{a}}_i$ using trainable action queries such as [16]. The BC loss of GC-ACT is the following $L1$ loss.

$$\mathcal{L}_{bc} = ||\hat{\mathbf{a}}_i - \mathbf{a}_i||_1 \tag{6}$$

The total loss for GC-ACT, auxiliary encoders and codebooks with pre-defined constants $c_1$ and $c_2$ is as follows.

$$\mathcal{L}_{GC\text{-}ACT} = \mathcal{L}_{bc} + c_1\mathcal{L}_h + c_2\mathcal{L}_z \tag{7}$$

### *B. Time-Optimization Transformer*

We propose TO-Transformer as the high-level policy to generate $(h_t^q, z_t^q)$, and predict an expected time cost for the generated $(h_t^q, z_t^q)$ to obtain fast imitation behavior. The inputs of TO-Transformer are current states $s_t$ and the trainable queries for the subgoal and style, and the outputs are the categorical distributions for the indices corresponding to the $h$ and $z$ codebooks. Then, $(h_t^q, z_t^q)$ referred by the predicted indices is used as autoregressive inputs for TO-Transformer, and the distribution of the expected time cost is finally predicted from the time-cost query.

For training TO-transformer, we sample $(s_i, \mathbf{a}_i, s_j, T_i)$ from the dataset where $T_i$ is the remained timesteps from $s_i$ to the end of the trajectory. Then, we obtain $(h_i^q, z_i^q)$ using the encoders and codebooks trained with GC-ACT. These encoders and codebooks are freezed while training TO-Transformer. The indices of $(h_i^q, z_i^q)$ are used as the supervised labels for the output of the categorical distributions, and we use a categorical cross-entropy loss $\mathcal{L}_{cce}$ to train the transformer. The loss of predicting the subgoal and style $\mathcal{L}_{latent}$ is as follows with the predefined constant $c_3$.

$$\mathcal{L}_{latent} = \mathcal{L}_{cce}(h_i^q, \hat{h}_i^q) + c_3\mathcal{L}_{cce}(z_i^q, \hat{z}_i^q) \tag{8}$$

Next, we train a probabilistic model to predict the expected time cost $\hat{T}_i$. We discretize the continuous $T_i$ by binning with the predefined size, and use a categorical cross-entropy loss to make the model predict the index of the bin. The loss for the time cost $\mathcal{L}_{time}$ is as follows.

$$\mathcal{L}_{time} = \mathcal{L}_{cce}(T_i, \hat{T}_i) \tag{9}$$

The total loss for TO-Transformer is as follows with the predefined cost $c_4$.

$$\mathcal{L}_{TO} = \mathcal{L}_{latent} + c_4\mathcal{L}_{time} \tag{10}$$

For inference, we generate several samples of $(h_t^q, z_t^q)$ from the given states and queries of the subgoal and style. Each of the sampled $(h_t^q, z_t^q)$ is used as the autoregressive inputs of TO-Transformer to predict the expected time cost. Then, we choose one of $(h_t^q, z_t^q)$ samples with the minimum expected time cost. The selected $(h_t^q, z_t^q)$ is provided to the low-level policy to finally obtain fast imitation behavior.

## V. Experiments

In this section, we provide experimental evidence to show that the proposed HACT-Vq improves imitation performance for the dataset with temporal multimodality. For that, we use both simulation and real-world environments. In simulation experiments, we generate datasets with scripted policies by controlling a speed factor so that the episode length of the datasets follows various distributions. These datasets are used to show that our model can learn demonstrations with the various types of temporal multimodality. In real-world experiments, we use teleoperation to collect demonstrations. As multiusers with different proficiency joined to collect datasets, the datasets have temporal multimodality. We show that the proposed HACT-Vq achieves the highest success rate compared to baselines.

### *A. System setup*

We build a dual-arm robot system with 16 degrees of freedom (DoF) based on Franka Emika Panda ($7 \times 2$ DoF) and grippers ($1 \times 2$ DoF). Two cameras are perched on the top of a workspace whose dimension is $640 \times 480$. We use the two RGB images $I_t \in \mathbb{R}^{640\times480\times3\times2}$ and joint states including gripper joints $q_t \in \mathbb{R}^{16}$ as the input states $s_t = (I_t, q_t)$ for our model. Practically, we only use $s_t = (I_t, )$ for the inputs of TO-Transformer because the images contain enough information for high-level decisions such as a subgoal and style. On the other hand, we use $s_t = (I_t, q_t)$ as inputs for GC-ACT to enhance the quality of the low-level policy by providing full observations. We use actions $a_t \in \mathbb{R}^{16}$ as the joint commands for the manipulator and gripper.

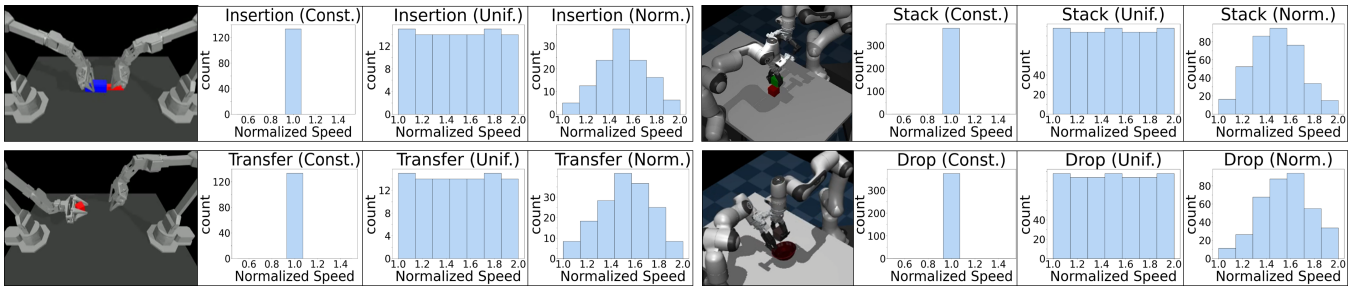


Fig. 3: Tasks in simulation experiments and distributions of the speeds of demonstrations in datasets.

| | **Cube Transfer (Sim [4])** | | | **Bimanual Insertion (Sim [4])** | | | **Cube Stack (Our sim)** | | | **Cube Drop (Our sim)** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Types of datasets | Const. | Unif. | Norm. | Const. | Unif. | Norm. | Const. | Unif. | Norm. | Const. | Unif. | Norm. |
| ACT | 70% | 20% | 60% | 70% | 40% | 60% | **100%** | 60% | 80% | **90%** | 60% | 20% |
| ACT-Vq | 30% | 20% | 70% | 50% | 0% | 30% | 90% | 80% | 80% | 80% | 60% | 20% |
| HACT | 40% | 30% | 30% | 0% | 0% | 0% | 90% | 90% | 70% | 80% | 20% | 20% |
| HACT-Vq (w/o TO) | **100%** | 30% | 70% | 50% | 40% | 30% | 90% | 70% | 20% | 30% | 10% | 20% |
| **(Ours)** HACT-Vq | **100%** | **80%** | **90%** | **80%** | **80%** | **80%** | 90% | **100%** | **100%** | **90%** | **70%** | **50%** |

TABLE I: Success rates of simulation tasks with three types of datasets. The speeds of demonstrations in a dataset are constant (*Const.*) or variable such as uniform distribution (*Unif.*) or normal distribution (*Norm.*).

### *B. Simulation experiments*

We use two types of simulation environments: the benchmark simulation proposed in [4] and our digital twin simulation. *Cube Transfer* and *Bimanual Insertion* tasks are adopted from the benchmark simulation, and *Cube Stack* and *Cube Drop* tasks are used in our simulation. For *Cube Stack*, one arm picks and places a red cube in the center, and the other arm picks and places a green cube on the red cube. For *Drop Cube*, one arm picks a basket, and the other arm picks and drops a cube in the basket. We randomized the location of the objects on a table through all episodes. We used script policies to collect demonstrations, 100 demonstrations for the benchmark simulation tasks, and 300 demonstrations for our simulation tasks. We intentionally modulated a speed factor of the script policies, and generated the three types of datasets which have different distributions of the speed. In Constant distribution (*Const.*), all the speeds of the demonstrations are the same. In Uniform distribution (*Unif.*), the speeds of the demonstrations are uniformly distributed. In Normal distribution (*Norm.*), the speeds of the demonstrations has a truncated normal distribution. Fig. 3 shows the snapshots of tasks and distributions of speed in the dataset.

We compared the success rates of the various models by training with the different types of the datasets. The experimental results are shown in TABLE I. *ACT* [4] is a baseline model which uses a temporal ensemble of action chunks and continuous style representations. *ACT-Vq* is the variation of *ACT* by adopting discrete style representations via Vq-VAE. We also compared *HACT* which is a hierarchical approach

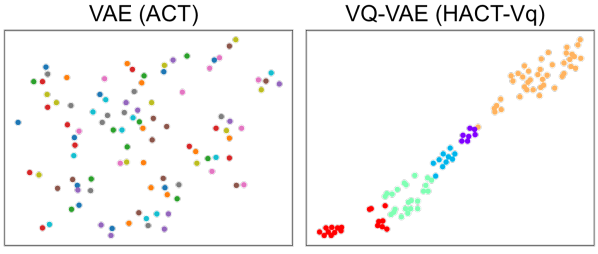


Fig. 4: t-SNE projection of style embeddings in VAE (ACT) and Vq-VAE (HACT-Vq).

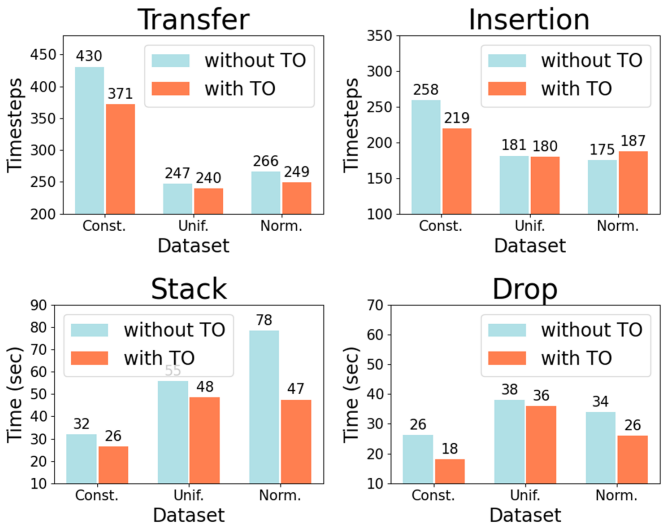


Fig. 5: Averaged time cost to finish a task with and without Time-Optimization (TO).

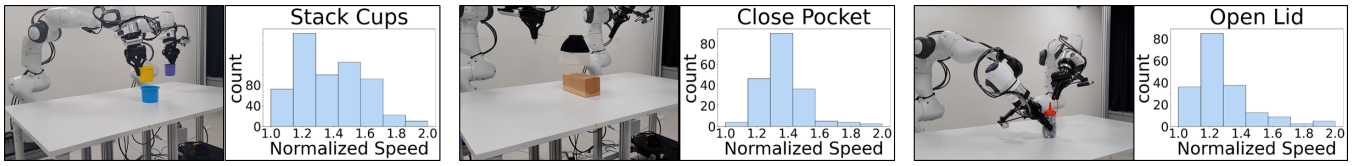


Fig. 6: Tasks in real-world experiments and distributions of speed of demonstrations in datasets.

where the low-level policy is GC-ACT with continuous subgoal and style representations, and the high-level policy is a simple transformer decoder trained as a regression model. Finally, we evaluated *HACT-Vq* which uses GC-ACT and To-Transformer with discrete subgoal and style representations as proposed in this paper. For *HACT-Vq (w/o TO)*, we did not use the subgoal and style based on the prediction of the time cost, but uses the argmax of the predicted indices of the subgoal and style.

In the results, the proposed *HACT-Vq* shows improved success rates in the most of tasks and datasets. For datasets without temporal multimodality (*Const.*), the baseline ACT showed good performance, and our proposed model slightly improved the success rates. The success rates of ACT, however, are significantly dropped in datasets with temporal multimodality such as *Unif.* and *Norm.*, On the other hand, HACT-Vq showed good performance in the datasets with various temporal modes. We consider that discrete embeddings via Vq-VAE can encode the representations better than continuous embeddings via VAE. Fig. 4 shows t-SNE projection of $z$ embeddings in VAE (ACT) and Vq-VAE (HACT-Vq). We found that VAE fails to learn meaningful representations of styles in that embeddings of similar styles are not clustered in the latent space. On the other hand, discrete representations in Vq-VAE encode similar styles into a similar space so that the mode of styles are discriminable in the latent space. Additionally, the discrete representations allow the high-level transformer to learn a probabilistic model for multimodality.

Next, we compared HACT-Vq with and without Time-Optimization (TO) to show that the proposed time-optimization provides the mode of fast imitation behavior. We measured the averaged time cost of episodes to achieve the task, and showed the result in Fig. 5. In the figure, HACT-Vq *with TO* shows the faster behavior compared to HACT-Vq *without TO*. In HACT-Vq *without TO*, the subgoal and style are decoded as those in the datset distribution, so all the suboptimal behaviors are mixed in. On the other hand, the proposed TO selectively decodes the subgoal and style with the mode of the minimum time cost to retrieve fast imitation behavior. It is also interesting that the proposed time-optimization not only reduces time cost but also increases success rates as shown in TABLE I. We consider that the mode of fast imitation is near-optimal behavior compared to the other slow behavior, and sampling and selection process in time-optimization makes the inference robust.

| | **Stack Cups** | **Close Pocket** | **Open Lid** |
|---|---|---|---|
| ACT | 10% | 50% | 60% |
| **(Ours) HACT-Vq** | **80%** | **70%** | **90%** |

TABLE II: Success rates of real-world tasks.

### C. Real-world experiments

In real-world experiments, we choose three tasks: *Stack Cups*, *Close Pocket*, *Open Lid*. For *Stack Cups*, the left arm picks up a yellow cup and stacks it on a blue cup, and the right arm picks up a purple cup and stacks it on the yellow cup. For *Close Pocket*, two arms collaborate to pick up a pocket and tie it up to close. For *Open Lid*, one arm grasps the sauce container, and the other arm opens its lid. We used teleoperation for collecting 300 demonstrations for *Stack Cups*, and 150 demonstrations for *Close Pocket* and *Open Lid*. We present the snapshots of each task and the distribution of speed in Fig. 6. Note that the speeds of the real-world tasks are various because the teleoperators with different proficiency joined.

TABLE II shows the results of the real-world experiments. In the results, our proposed model shows improved success rates among all tasks. Especially, we can observe that the success rates between ACT and ours are significantly different in *Stack Cups*. This task has the complex multimodality in both time and space in that proficient teleoperators pick two cups at once and stack them within a few seconds while less-skilled teleoperators pick and stack a cup one by one. This temporal multimodality causes the significant performance gap between ACT and the proposed model because the proposed model is robust to multimodality.

## CONCLUSION

We design a hierarchical policy for behavioral cloning of demonstrations with temporal multimodality. We propose Time-Optimization Transformer (TO-Transformer) as a high-level policy and Goal-Conditioned Action Chunking Transformer (GC-ACT) as a low-level policy. The proposed model is designed to learn the multimodality of demonstrations, especially various temporal modalities. For that, we use vector-quantization for the latent subgoal and style for the low-level policy via Vq-VAE. We also design time optimization for the high-level policy to retrieve fast imitation behavior via time-cost prediction. We apply the proposed model to bimanual tasks in real-world robots where demonstration data have various types of temporal modalities. The proposed model showed improved success rates with retrieving fast imitation behavior compared to baseline models.

## REFERENCES


[1] Anthony Brohan et al. "Rt-1: Robotics transformer for real-world control at scale". In: *arXiv preprint arXiv:2212.06817* (2022).

[2] Nur Muhammad Shafiullah et al. "Behavior Transformers: Cloning $k$ modes with one stone". In: *Advances in neural information processing systems* 35 (2022), pp. 22955–22968.

[3] Konstantinos Bousmalis et al. "RoboCat: A Self-Improving Foundation Agent for Robotic Manipulation". In: *arXiv preprint arXiv:2306.11706* (2023).

[4] Tony Z Zhao et al. "Learning fine-grained bimanual manipulation with low-cost hardware". In: *Robotics: Science and Systems Conference (RSS)* (2023).

[5] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. "Neural discrete representation learning". In: *arXiv preprint arXiv:1711.00937* (2017).

[6] Fang-I Hsiao, Jui-Hsuan Kuo, and Min Sun. "Learning a multi-modal policy via imitating demonstrations with mixed behaviors". In: *arXiv preprint arXiv:1903.10304* (2019).

[7] Karol Hausman et al. "Multi-modal imitation learning from unstructured demonstrations using generative adversarial nets". In: *Advances in neural information processing systems* 30 (2017).

[8] Cong Fei et al. "Triple-GAIL: a multi-modal imitation learning framework with generative adversarial nets". In: *arXiv preprint arXiv:2005.10622* (2020).

[9] Cheng Chi et al. "Diffusion policy: Visuomotor policy learning via action diffusion". In: *arXiv preprint arXiv:2303.04137* (2023).

[10] Octo Model Team et al. *Octo: An open-source generalist robot policy*. 2023.

[11] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[12] Scott Reed et al. "A generalist agent". In: *arXiv preprint arXiv:2205.06175* (2022).

[13] Zichen Jeff Cui et al. "From play to policy: Conditional behavior generation from uncurated robot data". In: *arXiv preprint arXiv:2210.10047* (2022).

[14] Jianrong Zhang et al. "T2m-gpt: Generating human motion from textual descriptions with discrete representations". In: *arXiv preprint arXiv:2301.06052* (2023).

[15] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[16] Nicolas Carion et al. "End-to-end object detection with transformers". In: *European conference on computer vision*. Springer. 2020, pp. 213–229.