A. Model Architecture Details

The Residual VQ-VAE model is a residual vector-quantized variational autoencoder designed for action sequence modeling. The architecture consists of an encoder, a residual vector quantization module, and a decoder, all implemented with causal convolutional layers to preserve temporal dependencies.

A.1. Encoder

The encoder takes as input action sequences with one channel and encodes them into a latent representation of dimension 128. It is composed of four blocks. Each block contains four residual layers, with output channels set to [128, 256, 256, 512]. The encoder uses the SiLU activation function and group normalization with 32 groups. The encoder output is further processed by a group normalization layer, a SiLU activation, and a final convolutional layer to produce the latent features.

A.2. Residual Vector Quantization

The latent features from the encoder are quantized using a Residual Vector Quantization (RVQ) module. The RVQ module uses 4 codebook groups, each with 256 entries, and an embedding dimension of 128. K-means initialization is used for the codebooks to stabilize training. The quantized latent vectors are then passed to the decoder.

A.3. Decoder

The decoder reconstructs the action sequence from the quantized latent representation. It consists of four blocks, with four layers per block and output channels [128, 256, 256, 512]. The decoder mirrors the encoder in terms of activation (SiLU) and normalization (group norm, 32 groups). The final output is projected to the original action dimension using a linear layer, followed by normalization, activation, and a final convolution. The output is sliced to match the original action window size.

B. Real-World Experiments Hardware Platform

In this section, we detail the specifics of our real-world hardware platform, as shown in Fig.4. Our setup primarily comprises a single Franka Research3 robotic arm paired with a third-person-view RealSense D435 camera. This camera is securely mounted in a fixed position, enabling it to capture comprehensive environmental observations with a resolution of 640x480 pixels. The entire system operates at a consistent frequency of 20 Hz. Actions for the robotic arm are precisely defined as absolute end-effector poses within the SE(3) space, which includes both the 3D position and the quaternion orientation. For data collection, we leveraged existing code from the Deoxys Control repository[1], utilizing a 3D mouse for teleoperation.

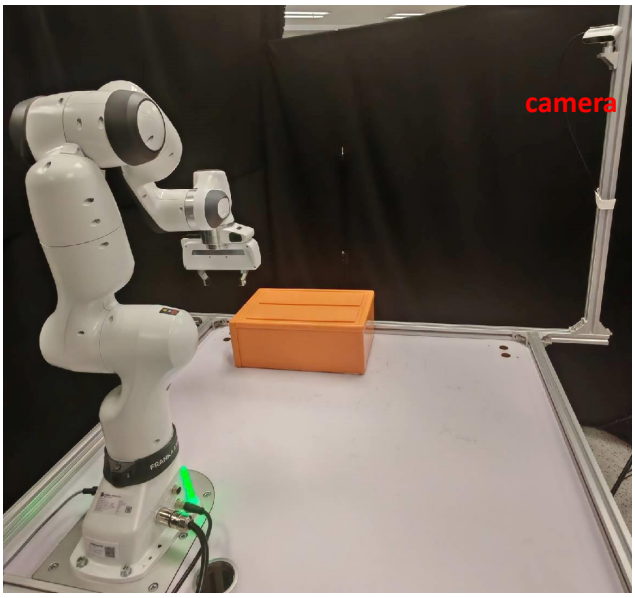

Figure 4. **The Real-world Franka Robot Arm Experiments Hardware Platform.**

C. The Evaluation Experiment Results.

C.1. The Success Rate Results of All Tasks.

We present the experimental results for three distinct models: VQ_{O}, VQ_{O+L}, and VQ_{O+L+M}. These models were evaluated on both the Libero-90 and a series of real-world experiments. The comprehensive results are summarized in Tab.7.

C.2. Comparison with Other Tokenizers

We compared our VQ_{O+L+M} action tokenizer with Fast[35] and Quest[32] on two real-world tasks: "pick the snake" and "put the snake into the drawer." All experiments utilized the OpenVLA backbone, and all tokenizers were deployed in a zero-shot manner without any fine-tuning (only the VLA backbone was fine-tuned). Results are in Tab.8

C.3. LIBERO-Long results

While LIBERO-90 already incorporates long-horizon tasks within its benchmark, we further evaluated our best tokenizer on LIBERO-Long. Our results demonstrated an improvement: VQ-VLA achieved a success rate of 55%, which represents a 4% increase over OpenVLA's 51% success rate on the same benchmark. This performance uplift underscores VQ-VLA's enhanced ability to manage and execute complex, multi-step tasks.

[1] https://github.com/UT-Austin-RPL/deoxys_control

	baseline(%)	VQ_O(%)	VQ_O+L(%)	VQ_O+L+M(%)
LIBERO-90	73.53	71.93	86.16	-
Pick up the [TOY NAME]	37	33	40	55
Put the toy into the basket	20	35	35	45
Flip the pot upright	30	45	45	60
Pull out a tissue paper	5	25	20	25
Short-horizon average	23	34.5	35	46.25
Put all cups in the basket	15	15	40	50
Put the toy into the drawer	5	15	15	30
Long-horizon average	10	15	25	40

Table 7. **The success rates for all tasks**

Tokenizer	pick the snake	put the snake into the drawer
FAST	20	5
QueST	10	0
VQ_{O+L+M} (Ours)	**65**	**20**

Table 8. **Success rate with different tokenizers.**

C.4. Decoder Capacity.

For fair comparison, all VLA backbones in our experiments were finetuned using the same data. Our VQ head adds negligible parameters (0.9% of total), indicating improvements stem from our modeling approach. Comparison with a scratch end-to-end action head on OpenVLA will be added in the camera-ready version. Additionally, we compared the VQ head with a scratch end-to-end action head on OpenVLA, as well as with UniAct[58]. For the scratch end-to-end action head, the final layer features from the VLA output were directly fed into a Multi-Layer Perceptron (MLP) to produce the action. Results are in Tab.9

	LIBERO-90(%)
UniAct	61.69
scratch end-to-end head	2.91
VQ head (Ours)	**86.16**

Table 9. **Success rate with different decoder capacities.**