Latent Action Pretraining from Videos

Ye et al. 2025

What the paper wants to tackle

- Foundation robot models (e.g. VLAs, WAMs) require high quality teleoperation data which is scarce.

- Internet video data is available in abundance but there are no labels and the distribution of the data significantly differs (e.g. embodiment, environment)

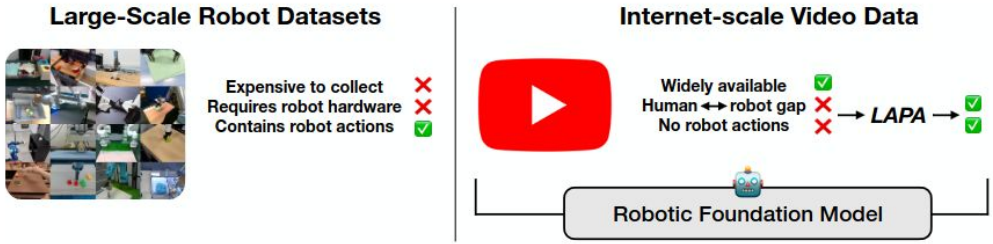


Figure 1: **Problem Formulation.** We investigate building a generalist robotic foundation model from human motion videos without action labels.

Model Description

Two models learned sequentially followed by a fine-tuning stage to map the latent actions to real robot actions.

- Genie style VQ-VAE
- VLM based on 7 Billion parameter Large World Model (LWM-chat-1M)

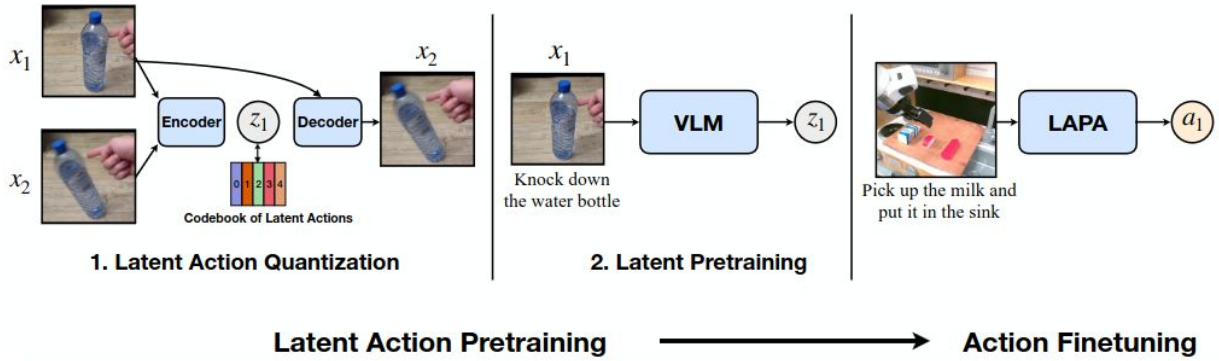


Figure 2: **Overview of Latent Action Pretraining**. (1) Latent Action Quantization: We first learn discrete latent actions in a fully unsupervised manner using the VQ-VAE objective (Detail in Figure 8). (2) Latent Pretraining: The VLM is trained to predict latent actions, essentially performing behavior cloning. After pretraining, we finetune LAPA on a small set of action-labeled trajectories to map the latent space to the end effector delta action space.

The VQ-VAE based Objective

- Learning quantized latent actions between raw image frames.
- Analogous to tokenizing atomic actions without the need to know action priors like end-effector positions or joint positions.
- LAPA captures environment-centric actions, including object and camera movement which can be beneficial for navigation, dynamic or non-quasistatic tasks (tasks requiring fast action).
- The VQ-VAE objective enables latent action z_t to be discrete tokens (codebooks), making it easy for VLA/VLMs to predict z_t.
- The latent action is represented using ´s´ sequences from |C| codebook vocabulary space.
- The VLA/VLM predicts latent actions using only one layer of MLP attached to the head.
- They used vocab of 8 and sequence length of 4 for the experiments.

Moving to actual robot action prediction

- Discretized continuous action space for each dimension of the robot (similar to OpenVLA)
- Latent action head replaced with a new action head to generate ground truth actions.
- Similar to latent pre-training, the vision encoder remains frozen and all the other parameters of the language model are unfrozen.

Encoder-Decoder Architecture

- Based on the the Genie 1 latent action model architecture
- Uses only two image frames the one before and the one after the action (unlike Genie because of computational limits)
- The quantization model is a variant of C-ViViT tokenizer as the encoder has both spatial and temporal components and the decoder only the spatial one (only two frames unlike Genie)
- The time distance between the two frames is H which is typically set to 0.6 seconds
- Instead of additive embeddings, cross attention is used to attend z_t to x_t (which leads to more semantically meaningful actions)

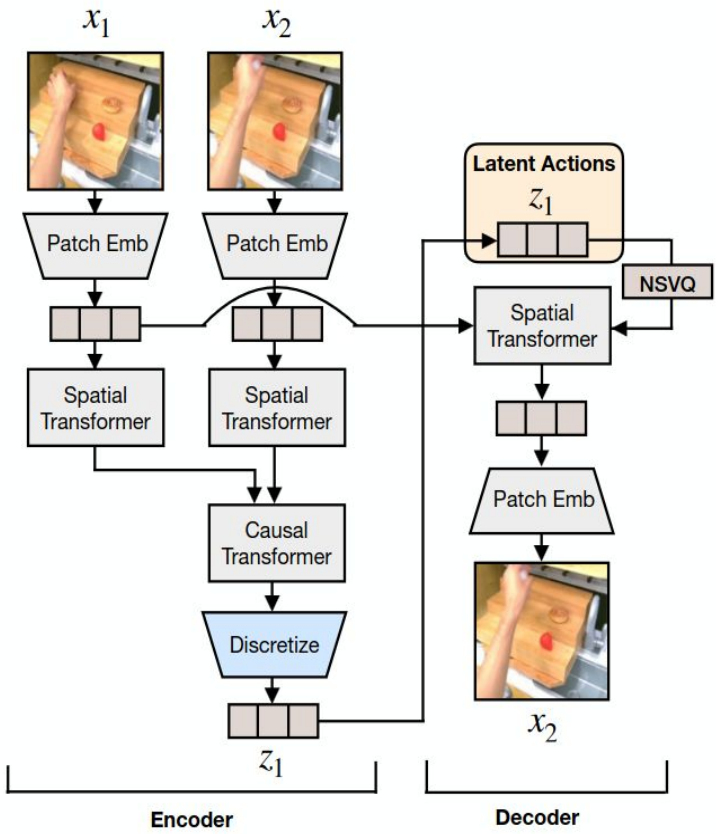


Figure 8: **Model architecture of our Latent Action Quantization Model.**

Comparisons: Language Table

- Scratch: No video training (only fine-tuning the VLM)
- UniPi: WAM which also gets fine-tuned on the downstream task + IDM
- VPT: IDM which gets used to train LAPA as an alternative to latent actions
- ActionVLA: LAPA trained on ground truth answers.
- OpenVLA: fine-tuned on downstream tasks

Table 1: **Language Table Results.** Average Success Rate (%) ± StdErr across the three different pretrain-finetune combinations from the Language Table benchmark as described in Table 3. We also note the # of trajectories used for fine-tuning next to each category.

	In-domain (1k)		Cross-task (7k)		Cross-env (1k)	
	Seen	Unseen	Seen	Unseen	Seen	Unseen
SCRATCH	$15.6_{\pm 9.2}$	$15.2_{\pm 8.3}$	$27.2_{\pm 13.6}$	$22.4_{\pm 11.0}$	$15.6_{\pm 9.2}$	$15.2_{\pm 8.3}$
UNIPI	$22.0_{\pm 12.5}$	$13.2_{\pm 7.7}$	$20.8_{\pm 12.0}$	$16.0_{\pm 9.1}$	$13.6_{\pm 8.6}$	$12.0_{\pm 7.5}$
VPT	$44.0_{\pm 7.5}$	$32.8_{\pm 4.6}$	$72.0_{\pm 6.8}$	$\mathbf{60.8}_{\pm 6.6}$	$18.0_{\pm 7.7}$	$18.4_{\pm 9.7}$
LAPA	$\mathbf{62.0}_{\pm 8.7}$	$\mathbf{49.6}_{\pm 9.5}$	$\mathbf{73.2}_{\pm 6.8}$	$54.8_{\pm 9.1}$	$\mathbf{33.6}_{\pm 12.7}$	$\mathbf{29.6}_{\pm 12.0}$
ACTIONVLA	$77.0_{\pm 3.5}$	$58.8_{\pm 6.6}$	$77.0_{\pm 3.5}$	$58.8_{\pm 6.6}$	$64.8_{\pm 5.2}$	$54.0_{\pm 7.0}$

Comparisons: OpenX and Bridge

OpenVLA trained from scratch for Bridge and taken from the original checkpoints for the OpenX version.

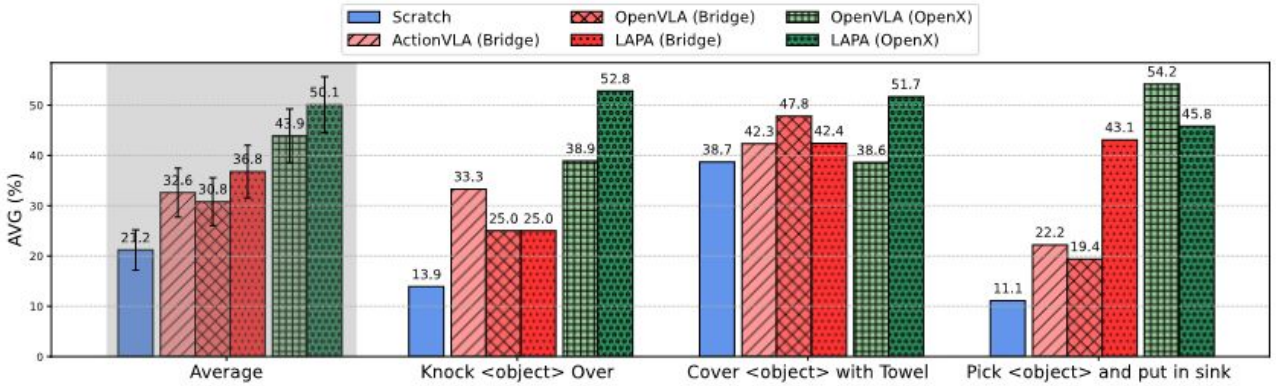
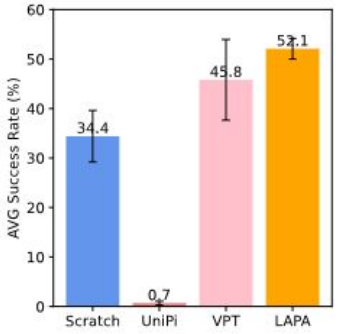


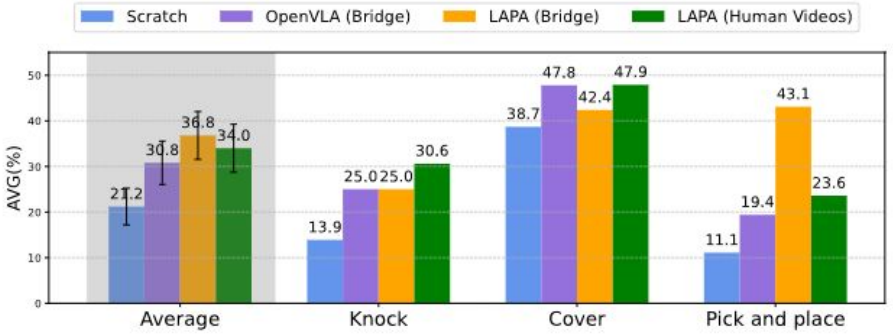
Figure 3: **Real-world Tabletop Manipulation Results.** We evaluate on a total of 54 rollouts for each model encompassing unseen object combinations, unseen objects and unseen instructions. Average success rate (%) ± StdErr are shown (detailed results provided in Appendix G.3).

Training on

The fine-tuning data is what does the embodiment grounding, and it's small, which is exactly why grasping (the most fine-grained, embodiment-specific skill) is LAPA's weakest point — 450 trajectories aren't enough to nail precise grasps, whereas the coarse "reach toward the right object" planning that came from pretraining is strong.



(a) SIMPLER Results



(b) Real-world Tabletop Manipulation Robot Results

Conclusion

- Trained a VQ-VAE to go from encoding to latent action and from latent action and the frame input to the next frame using the decoder; meaning:
  - The encoder is a latent Inverse Dynamics
  - The decoder is a Dynamics model (a.k.a world model)
- Allows model to learn from any video and allows for better embodiment transfer as the latent actions are more abstract than robot actions.