Latent Action Pretraining From Videos

Anonymous Author(s)
Affiliation
Address
email

Abstract: We introduce Latent Action Pretraining for general Action models (LAPA), the first unsupervised method for pretraining Vision-Language-Action (VLA) models without ground-truth robot action labels. Existing Vision-Language-Action models require action labels typically collected by human teleoperators during pretraining, which significantly limits possible data sources and scale. In this work, we propose a method to learn from internet-scale videos that do not have robot action labels. We first train an action quantization model leveraging VQ-VAE-based objective to learn discrete latent actions between image frames, then pretrain a *latent* VLA model to predict these latent actions from observations and task descriptions, and finally finetune the VLA on small-scale robot manipulation data to map from latent to robot actions. Experimental results demonstrate that our method outperforms the state-of-the-art VLA model trained with robotic action labels on real-world manipulation tasks that require language conditioning, generalization to unseen objects, and semantic generalization to unseen instructions. Training only on human manipulation videos also shows positive transfer, opening up the potential for leveraging web-scale data for robotics foundation model.

Keywords: Vision-Language-Action Models, Unsupervised Learning

1 Introduction

Vision-Language-Action Models (VLA) for robotics [1, 2] are trained by aligning large language models with vision encoders, and then finetuning it on on diverse robot datasets [3]; this enables generalization to novel instructions, unseen objects, and distribution shifts [4]. However, diverse real-world robot datasets mostly require human teleoperation, which makes scaling difficult. Internet video data, on the other hand, offers abundant examples of human behavior and physical interactions at scale, presenting a promising approach to overcome the limitations of small, specialized robotic datasets [5]. However, it is challenging to learn from internet video data for two major challenges: first, much of the raw data on the web lacks explicit action labels; second, the data distribution from the web is fundamentally different from the embodiments and environments of typical robotic systems [6]. We propose **L**atent **A**ction **P**retraining for General **A**ction Models (LAPA), an unsupervised approach to pretraining a robotic foundation model without the need for ground-truth robot action labels.

LAPA has two pretraining stages, followed by a fine-tuning stage to map the latent actions to real robot actions. In the first pretraining stage, we use a VQ-VAE-based objective [7] to learn quantized latent actions between raw image frames. Analogous to Byte Pair Encoding [8] used for language modeling, this can be seen as learning to tokenize atomic actions without requiring predefined action priors (e.g., end-effector positions, joint positions). In the second stage, we perform behavior cloning by pretraining a Vision-Language Model to predict latent actions derived from the first stage based on video observations and task descriptions. Finally, we fine-tune the model on a small-scale robot

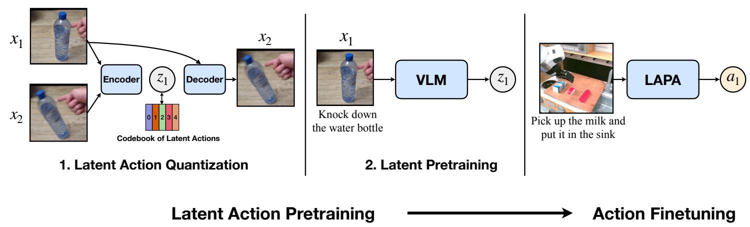


Figure 1: **Overview of LAPA**. (1) Latent Action Quantization: We first learn discrete latent actions in a fully unsupervised manner using the VQ-VAE objective. (2) Latent Pretraining: The VLM is trained to predict latent actions, essentially performing behavior cloning. After pretraining, we finetune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

manipulation dataset with robot actions to learn the mapping from the latent actions to robot actions. In this work, we refer to both the proposed method and the resulting VLA models as LAPA.

We measure performance on diverse manipulation videos, including existing robot video datasets (without utilizing ground-truth actions) and human manipulation datasets. On real-world manipulation tasks, our method leads to a new monolithic VLA model, outperforming OpenVLA, the current state-of-the-art model Vision Language Action (VLA) model trained on a diverse mixture of datasets with ground-truth actions. These results demonstrate the effectiveness of learning unified quantized latent action representations across diverse robotic datasets featuring different embodiments (shown in Appendix C). We further demonstrate that LAPA remains effective even when pretrained on *only* human manipulation video, outperforming models pretrained on Bridgev2, one of the largest open-sourced robotic datasets. We expect that our method opens up the potential for building foundation models for robotics by pretraining on much larger web-scale video data.

Our main contributions and findings are as follows: (1) We propose Latent Action Pretraining for general Action models (LAPA), an unsupervised approach to pretraining a robotic foundation model to encode robotic skills from web-scale video data. (2) Experiments on simulation and real-world robot tasks show that our method not only significantly outperforms baseline methods for training robotic manipulation policies from actionless video, but also leads to a VLA model that outperforms the current state-of-the-art VLA model trained with ground-truth actions (by +6.22%), while achieving over 30x greater pretraining efficiency.

2 LAPA: Latent Action Pretraining for general Action models

LAPA is divided into two stages: Latent Action Quantization and Latent Pretraining (Figure 1).

2.1 Latent Action Quantization

To learn latent actions in a fully unsupervised manner, we train a latent action quantization model following Bruce et al. [9] with a few modifications. Our latent action quantization model is an encoder-decoder architecture where the encoder takes the current frame x_t and the future frame x_{t+H} of a video with a fixed window size H and outputs the latent action z_t. The decoder is trained to take the latent action z_t and x_t and reconstruct x_{t+H}. Unlike Bruce et al. [9], we use cross attention to attend z_t given x_t instead of additive embedding, which empirically leads to capturing more semantically meaningful latent actions. Our quantization model is a variant of C-ViViT tokenizer [10] where the encoder includes both spatial and temporal transformer while the decoder only contains spatial transformer since our model uses only two image frames as input. Further model details are provided in Appendix G. Our latent action quantization training model is based on the VQ-VAE objective [11]. The VQ-VAE objective enables the latent action z_t to be discrete tokens (codebooks), making it easy for VLMs to predict z_t. The latent action is represented using s sequences from $|C|$ codebook vocabulary space. To avoid gradient collapse often observed in VQ-VAE, we utilize NSVQ [12] which replaces the vector quantization error to a product of original

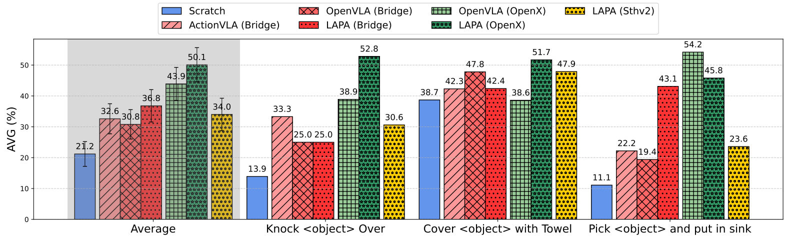


Figure 2: **Real-world Tabletop Manipulation Results.** We evaluate on a total of 54 rollouts for each model encompassing unseen object combinations, unseen objects and unseen instructions. Average success rate (%) are shown (detailed results provided in Appendix L.3).

error and a normalized noise vector. We also apply codebook replacement technique from NSVQ during early training steps to maximize codebook utilization.

2.2 Latent Pretraining

We use the encoder of the latent action quantization model as an inverse dynamics model to label all x_t, given x_{t+1}, with z_t. Then, we pretrain a VLM to predict the z_t given the language instruction of a video clip and the current image x_t. Instead of using the existing language model head of the VLM, we attach a separate latent action head of vocab size $|C|$. By default, we freeze only the vision encoder and unfreeze the language model during training. Since latent pretraining does not rely on ground truth actions, it opens the possibility of using any type of raw video paired with language instructions. Also, in contrast to traditional action granularity used in robotics (e.g. end-effector positions, joint positions, joint torques, etc.), our approach does not require any priors about the action hierarchy/granularity.

2.3 Action Finetuning

VLAs that are pretrained to predict latent actions are not directly executable on real-world robots since latent actions are not actual delta end-effector actions or joint actions. To map latent actions to actual robot actions, we finetune LAPA on a small set of labeled trajectories that contain ground truth actions (delta end-effector). For action prediction, we discretize the continuous action space for each dimension of the robot so that the number of data points allocated for each bin is equal following Kim et al. [2], Brohan et al. [1]. We discard the latent action head (a single MLP layer) and replace it with a new action head to generate ground truth actions. As with latent pretraining, we freeze the vision encoder and unfreeze all of the parameters of the underlying language model.

3 Experiments

In this section, we demonstrate the effectiveness of LAPA as a general-purpose pretaining method. Specifically, we focus on answering the following questions through a real-world tabletop manipulation setting: **Q1.** How does LAPA perform when there are cross-embodiment gaps between pretraining and fine-tuning? **Q2.** Can LAPA learn superior priors compared to using ground-truth actions during pretraining in a multi-embodiment setting? **Q3.** Can we create a performant LAPA solely from raw human manipulation videos? We provide details of experimental setups and baseline models in Appendix H and I. We also provide preliminary experiment results that compare the effect of LAPA with baseline methods of training manipulation policies from actionless videos on Language Table [13] and SIMPLER [14] in Appendix A and analysis regarding the scaling of LAPA in Appendix E.

We pretrain our models on (1) Bridgev2 for **cross-embodiment** performance (WidowX to Franka embodiment) , (2) Open X-Embodiment Dataset [3] to measure the effect of pretraining in a **multi-embodiment** setting and (3) Something-Something V2 dataset [15] to see the potential of LAPA pretrained on **human manipulation videos**. Figure 2 shows the average success rate across the 3 tasks where each task encompasses unseen object combination, object, and instruction settings. We provide detailed results depending on the generalization type in Table 12 in Appendix L.3.

Bridgev2 Pretraining We compare models that were pretrained on the Bridgev2 dataset. Similar to previous results, all models pretrained on Bridgev2 result in significant performance enhancement compared to SCRATCH. Furthermore, by comparing LAPA which does not leverage action-labeled trajectories during pretraining with models that use action-labeled trajectories during pretraining (ACTIONVLA and OPENVLA), we observe an interesting finding: LAPA outperform VLAs that use action labeled pretraining data on average success rate of the 3 tasks, unlike previous scenarios where VLAs pretrained on the ground-truth actions were upper bounds. LAPA significantly outperforms the other models in pick-and-place tasks; given that most tasks in Bridgev2 are pick-and-place, we hypothesize that VLA models pretrained on ground truth action labels have overfitted to the WidowX action space from the Bridgev2 dataset, hampering cross-embodiment adaptability to action distribution shifts during fine-tuning. In contrast, LAPA avoids this issue by not relying on ground truth action labels during pretraining.

Open-X Pretraining From Figure 2, we see that VLAs pretrained on the Open-X dataset outperforms VLAs pretrained on the Bridgev2 dataset, showing that data scaling during pretraining demonstrates positive transfer for downstream tasks [3]. This also suggests there could be significant further improvement when scaling the diversity and scale of the pretraining data, especially with large web-scale video data. When comparing LAPA with OPENVLA, we see that LAPA significantly outperforms OPENVLA on 2 out of 3 tasks (Figure 2). This highlights LAPA's effectiveness in a multi-embodiment setting by showcasing its ability to leverage a shared latent action space during pretraining, akin to how language and image representations are utilized. In contrast, contemporary action pretraining methods may suffer from reduced positive transfer between datasets due to the variability in action representation spaces across different embodiments and datasets. However, for pick and place task, LAPA underperforms OPENVLA. We observe that most failures of LAPA are due to early grasping. In fact, LAPA outperforms OPENVLA in reaching performance (83.33% vs 66.67%) (Appendix L.3). This suggests that, although LAPA possesses stronger language conditioning, there is room for improvement in skills such as grasping. Since grasping occurs only once or twice in each trajectory, the 150 labeled trajectories may not be sufficient for LAPA to accurately predict grasp actions based on the physical characteristics of diverse objects.

Human Video Pretraining We report the real-world robot experiments in Figure 2. Surprisingly, we can see that LAPA trained with human videos outperforms OPENVLA (Bridge) on average. Despite the larger embodiment gap for LAPA (Human to robot vs. Robot to robot), it learns a better prior for robot manipulation. This result highlights the potential of raw human manipulation videos from the web compared to expensive robot manipulation data, which requires time-intensive teleoperation to collect. Results comparing LAPA (Sthv2) with baseline models also trained with human video are shown in Appendix A.3.

4 Conclusion

In this paper, we introduce LAPA, a scalable pretraining method for building VLAs using actionless videos. Across three benchmarks spanning both simulation and real-world robot experiments, we show that our method significantly improves transfer to downstream tasks compared to existing approaches. We also present a state-of-the-art VLA model that surpasses current models trained on 970K action-labeled trajectories. Furthermore, we demonstrate that LAPA can be applied purely on human manipulation videos, where explicit action information is absent, and the embodiment gap is substantial. We also show the pretraining efficiency of LAPA in Appendix B and qualitative analysis in Appendix C. We believe our work can be extended to build scalable robot foundation models.

References

[1] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[2] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[3] O.-E. Collaboration, A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

[4] Z. Michał, C. William, P. Karl, M. Oier, F. Chelsea, and L. Sergey. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024.

[5] S. Yang, J. C. Walker, J. Parker-Holder, Y. Du, J. Bruce, A. Barreto, P. Abbeel, and D. Schuurmans. Position: Video as the new language for real-world decision making. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[6] R. McCarthy, D. C. Tan, D. Schmidt, F. Acero, N. Herr, Y. Du, T. G. Thuruthel, and Z. Li. Towards generalist robot learning from internet video: A survey. *arXiv preprint arXiv:2404.19664*, 2024.

[7] A. Van Den Oord, O. Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[8] R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016.

[9] J. Bruce, M. D. Dennis, A. Edwards, J. Parker-Holder, Y. Shi, E. Hughes, M. Lai, A. Mavalankar, R. Steigerwald, C. Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

[10] R. Villegas, M. Babaeizadeh, P.-J. Kindermans, H. Moraldo, H. Zhang, M. T. Saffar, S. Castro, J. Kunze, and D. Erhan. Phenaki: Variable length video generation from open domain textual descriptions. In *International Conference on Learning Representations*, 2023.

[11] A. van den Oord, O. Vinyals, and k. kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, 2017.

[12] M. H. Vali and T. Bäckström. Nsvq: Noise substitution in vector quantization for machine learning. *IEEE Access*, 10:13598–13610, 2022. doi:10.1109/ACCESS.2022.3147670.

[13] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, pages 1–8, 2023. doi:10.1109/LRA.2023.3295255.

[14] X. Li, K. Hsu, J. Gu, K. Pertsch, O. Mees, H. R. Walke, C. Fu, I. Lunawat, I. Sieh, S. Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.

[15] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, 2017.

[16] Y. Du, S. Yang, P. Florence, F. Xia, A. Wahid, brian ichter, P. Sermanet, T. Yu, P. Abbeel, J. B. Tenenbaum, L. P. Kaelbling, A. Zeng, and J. Tompson. Video language planning. In *The Twelfth International Conference on Learning Representations*, 2024.

[17] H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, 2023.

[18] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[19] C. Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

[20] H. Liu, W. Yan, M. Zaharia, and P. Abbeel. World model on million-length video and language with ringattention. *arXiv preprint arXiv:2402.08268*, 2024.

[21] M. Abdin, S. A. Jacobs, A. A. Awan, J. Aneja, A. Awadallah, H. Awadalla, N. Bach, A. Bahree, A. Bakhtiari, H. Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

[22] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[23] D. Niu, Y. Sharma, G. Biamby, J. Quenum, Y. Bai, B. Shi, T. Darrell, and R. Herzig. Llarva: Vision-action instruction tuning enhances robot learning. *arXiv preprint arXiv:2406.11815*, 2024.

[24] X. Li, C. Mata, J. Park, K. Kahatapitiya, Y. S. Jang, J. Shang, K. Ranasinghe, R. Burgert, M. Cai, Y. J. Lee, et al. Llara: Supercharging robot learning data for vision-language policy. *arXiv preprint arXiv:2406.20095*, 2024.

[25] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[26] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[27] S. Dasari, M. K. Srirama, U. Jain, and A. Gupta. An unbiased look at datasets for visuo-motor pre-training. In *Conference on Robot Learning*, 2023.

[28] H. Wu, Y. Jing, C. Cheang, G. Chen, J. Xu, X. Li, M. Liu, H. Li, and T. Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *The Twelfth International Conference on Learning Representations*, 2024.

[29] J. Liang, R. Liu, E. Ozguroglu, S. Sudhakar, A. Dave, P. Tokmakov, S. Song, and C. Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. *arXiv preprint arXiv:2406.16862*, 2024.

[30] J. Zeng, Q. Bu, B. Wang, W. Xia, L. Chen, H. Dong, H. Song, D. Wang, D. Hu, P. Luo, et al. Learning manipulation by predicting interaction. *arXiv preprint arXiv:2406.00439*, 2024.

[31] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[32] A. Kannan, K. Shaw, S. Bahl, P. Mannam, and D. Pathak. Deft: Dexterous fine-tuning for real-world hand policies. *arXiv preprint arXiv:2310.19797*, 2023.

[33] M. K. Srirama, S. Dasari, S. Bahl, and A. Gupta. Hrp: Human affordances for robotic pre-training. *arXiv preprint arXiv:2407.18911*, 2024.

[34] K. Shaw, S. Bahl, and D. Pathak. Videodex: Learning dexterity from internet videos. In *Conference on Robot Learning*, 2023.

[35] C. Wen, X. Lin, J. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023.

[36] H. Bharadhwaj, R. Mottaghi, A. Gupta, and S. Tulsiani. Track2act: Predicting point tracks from internet videos enables diverse zero-shot robot manipulation. *arXiv preprint arXiv:2405.01527*, 2024.

[37] C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023.

[38] Y. Zhu, A. Lim, P. Stone, and Y. Zhu. Vision-based manipulation from single human video with open-world object graphs. *arXiv preprint arXiv:2405.20321*, 2024.

[39] H. Bharadhwaj, A. Gupta, S. Tulsiani, and V. Kumar. Zero-shot robot manipulation from passive human videos. *arXiv preprint arXiv:2302.02011*, 2023.

[40] J. Ye, J. Wang, B. Huang, Y. Qin, and X. Wang. Learning continuous grasping function with a dexterous hand from human demonstrations. *IEEE Robotics and Automation Letters*, 8(5): 2882–2889, 2023.

[41] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. In *European Conference on Computer Vision*, 2022.

[42] J. Yang, Z.-a. Cao, C. Deng, R. Antonova, S. Song, and J. Bohg. Equibot: Sim (3)-equivariant diffusion policy for generalizable and data efficient learning. *arXiv preprint arXiv:2407.01479*, 2024.

[43] Y. Du, S. Yang, B. Dai, H. Dai, O. Nachum, J. B. Tenenbaum, D. Schuurmans, and P. Abbeel. Learning universal policies via text-guided video generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[44] P.-C. Ko, J. Mao, Y. Du, S.-H. Sun, and J. B. Tenenbaum. Learning to act from actionless videos through dense correspondences. In *The Twelfth International Conference on Learning Representations*, 2024.

[45] S. Yang, Y. Du, S. K. S. Ghasemipour, J. Tompson, L. P. Kaelbling, D. Schuurmans, and P. Abbeel. Learning interactive real-world simulators. In *The Twelfth International Conference on Learning Representations*, 2024.

[46] H. Bharadhwaj, D. Dwibedi, A. Gupta, S. Tulsiani, C. Doersch, T. Xiao, D. Shah, F. Xia, D. Sadigh, and S. Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. *arXiv preprint arXiv:2409.16283*, 2024.

[47] B. Baker, I. Akkaya, P. Zhokov, J. Huizinga, J. Tang, A. Ecoffet, B. Houghton, R. Sampedro, and J. Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. In *Advances in Neural Information Processing Systems*, 2022.

[48] A. D. Edwards, H. Sahni, Y. Schroecker, and C. L. Isbell. Imitating latent policies from observation. *arXiv preprint arXiv:1805.07914*, 2018.

[49] D. Schmidt and M. Jiang. Learning to act without actions. In *The Twelfth International Conference on Learning Representations*, 2024.

[50] K. Cobbe, C. Hesse, J. Hilton, and J. Schulman. Leveraging procedural generation to benchmark reinforcement learning. *arXiv preprint arXiv:1912.01588*, 2019.

[51] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet. Learning latent plans from play. In *Conference on robot learning*, pages 1113–1132. PMLR, 2020.

[52] Z. Jiang, Y. Xu, N. Wagener, Y. Luo, M. Janner, E. Grefenstette, T. Rocktäschel, and Y. Tian. H-gap: Humanoid control with a generalist planner. *arXiv preprint arXiv:2312.02682*, 2023.

[53] S. Lee, Y. Wang, H. Etukuru, H. J. Kim, N. M. M. Shafiullah, and L. Pinto. Behavior generation with latent actions. *arXiv preprint arXiv:2403.03181*, 2024.

[54] A. Mete, H. Xue, A. Wilcox, Y. Chen, and A. Garg. Quest: Self-supervised skill abstractions for learning continuous control. *arXiv preprint arXiv:2407.15840*, 2024.

[55] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[56] R. Villegas, M. Babaeizadeh, P.-J. Kindermans, H. Moraldo, H. Zhang, M. T. Saffar, S. Castro, J. Kunze, and D. Erhan. Phenaki: Variable length video generation from open domain textual descriptions. In *International Conference on Learning Representations*, 2022.

[57] S. Belkhale, T. Ding, T. Xiao, P. Sermanet, Q. Vuong, J. Tompson, Y. Chebotar, D. Dwibedi, and D. Sadigh. Rt-h: Action hierarchies using language. *arXiv preprint arXiv:2403.01823*, 2024.

[58] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[59] H. Kress-Gazit, K. Hashimoto, N. Kuppuswamy, P. Shah, P. Horgan, G. Richardson, S. Feng, and B. Burchfiel. Robot learning as an empirical science: Best practices for policy evaluation. *arXiv preprint arXiv:2409.09491*, 2024.

A Preliminary Experiments

A.1 Language Table Results

Table 1: **Language Table Results.** Average Success Rate (%) across the three different pretrain-finetune combinations from the Language Table benchmark as described in Table 2. 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	$77.0_{\pm 3.5}$	$58.8_{\pm 6.6}$	$77.0_{\pm 3.5}$	$58.8_{\pm 6.6}$	$64.8_{\pm 5.2}$	$54.0_{\pm 7.0}$

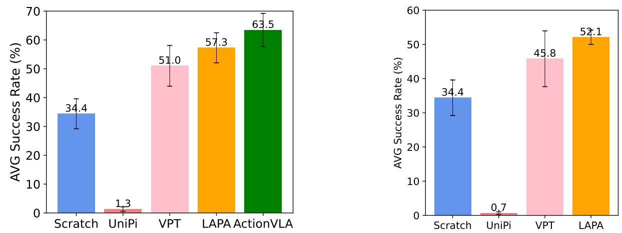
In-Domain Performance First, we assess LAPA's ability to learn from a small subset of in-domain action label data by pretraining on 181k trajectories and finetuning on 1k action-labeled trajectories (0.5%). As shown in Table 1, LAPA largely outperforms SCRATCH and narrows the gap with ACTIONVLA despite not using action labels during pretraining. Additionally, LAPA surpasses UNIPI and VPT. Notably, while UNIPI handles simple tasks well, its diffusion model often generates incorrect plans for longer-horizon tasks, aligning with Du et al. [16]. VPT, with the same backbone VLM as LAPA, outperforms UNIPI, showing the superiority of the VLA model, but still underperforms LAPA, highlighting the effectiveness of latent actions.

Cross-Task Performance We investigate whether LAPA's broad skills can be retained after fine-tuning on a specific task. Pretraining LAPA on 181k trajectories and finetuning on only separate tasks (7k), we evaluate all 5 task categories, similar to the in-domain setup, to assess latent pretraining's benefits for unseen tasks. When comparing LAPA and SCRATCH in Table 1 and Table 6, 7 in Appendix L.1, latent pretraining significantly benefits the separate task as well the other 4 task categories, resulting in a significant boost in both seen and unseen setups. Like before, UNIPI is constrained by its diffusion model's planning limitations, while VPT performs strongly, even surpassing ACTIONVLA in the unseen setting. This is likely due to using more labeled data (7k vs. 1k), helping the IDM generate more accurate pseudo labels.

Cross-Environment Performance We further investigate if LAPA benefits downstream performance when the pretraining and fine-tuning environments are different. We pretrain LAPA on 440k real-world trajectories, and then finetune on 1k simulation trajectories, which can be seen as testing on a setup where a real2sim gap is present (Figure 7 (a)). From Table 1, we observe that LAPA still significantly outperforms SCRATCH, showing that latent pretraining leads to positive transfer even on cross-environment setting. Notably, both UNIPI and VPT significantly underperforms LAPA, showing that learning to predict latent actions is more robust to cross-environment transfer. VPT only results in minor positive transfer, indicating that the IDM is not robust to environment shifts.

A.2 SIMPLER Results

We pretrain our models on the Bridgev2 [17] dataset and fine-tune on 100 trajectories collected from the SIMPLER environment [14]. As shown in Figure 3a, UNIPI significantly underperforms all other baselines on the SIMPLER Environment. We observe that, although the generated plans from the diffusion models are quite accurate, the IDM lacks the capability to predict 7 DOF continuous actions accurately when given only 100 action-labeled trajectories. This implies the effectivness of using VLAs in scenarios with insufficient action-labeled data. Similar to the results of Section A.1, LAPA outperforms baseline models that pretrain on actionless videos (UNIPI and VPT) and closes the performance gap with ACTIONVLA, which is pretrained on all of the 60K action-labeled trajectories from the Bridgev2 dataset. This highlights the effectiveness of LAPA, even when the complexity of the action space increases.



(a) Bridge to SIMPLER (b) Human to SIMPLER

Figure 3: **SIMPLER Results.** Average success rate (%) of LAPA and baselines pretrained on bridge and fine-tuned on SIMPLER (left). We also pretrain on human manipluation videos where the embodiment and environment gap is extreme and fine-tune on SIMPLER (right).

A.3 Human Manipulation Videos

We first evaluate the performance of LAPA pretrained on human videos on SIMPLER. In addition to SCRATCH, we also compare with UNIPI and VPT pretrained with the same human video dataset. As shown in Figure 3b, LAPA outperforms SCRATCH, showing that although the distribution of the pretraining data is distinct from the deployment setup, leveraging human videos for latent action pretraining results in positive transfer. Also, consistent with the result of Section A.2, LAPA shows the best performance, implying that Latent Action Pretraining is robust to human to robot embodiment shifts. Note that it is impossible to train ACTIONVLA because the human videos do not have any robot action labels.

B Pretraining Efficiency

The benefit of LAPA extends beyond downstream task performance to include pretraining efficiency. For pretraining LAPA (Open-X), the best-performing model, we use 8 H100 GPUs for 34 hours with a batch size of 128 (total of 272 H100-hours). In contrast, OPENVLA required a total of 21,500 A100-hours with a batch size of 2048. Despite being approximately 30-40 times more efficient for pretraining, LAPA still outperforms OPENVLA. We believe this efficiency stems from two factors. First, the training objective during LWM pretraining which corresponds to generating the next frame in a video, enables the model to implicitly understand high-level actions in a video. Notably, ACTIONVLA (Bridge), which uses LWM as the backbone reaches optimal performance in significantly fewer epochs (3 epochs) compared to OPENVLA (Bridge), which uses Prismatic as the backbone (30 epochs). Second, the action space for LAPA is much smaller than that for OPENVLA (8^4 vs. 256^7), making learning the perception-and-language to action generation problem easier to learn. For all LAPA models (BridgeV2, Open-X, Human Videos), we observe that a single epoch of training is sufficient to achieve optimal performance.

C Latent Action Analysis

We qualitatively analyze the alignment of quantized latent actions with real continuous actions. For interpretation, we condition the current image observation x_1 and each latent action on the decoder of the latent action quantization model, and present the reconstructed images. In Language Table, we observe that each latent action corresponds to a distinct movement of the robot arm (shown in Figure 11, 12 of Appendix K). Next, for human manipulation videos, we observe that camera viewpoints also correspond to a latent action (shown in Figure 13 of Appendix K). We also analyze the latent actions learned from the Open-X embodiment, which encompasses multiple embodiments, tasks, and environments. As shown in Figure 4, even though the embodiment and environment differ, conditioning on the same latent action results in a similar action in the reconstructed image. This supports our previous claim that latent actions are learned in a shared representation space, facilitating stronger positive transfer across diverse datasets.

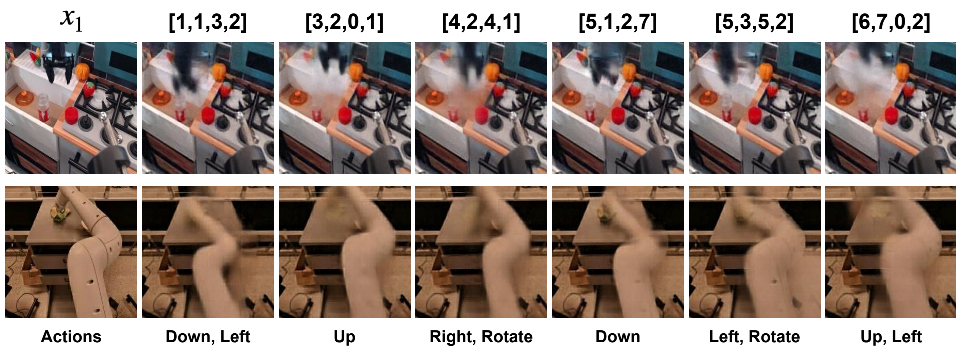


Figure 4: **Latent Action Analysis.** We condition the current observation x_1 and quantized latent action to the decoder of the latent action quantization model. We observe that each latent action can be mapped into a semantic action. For example, latent action [1,1,3,2] corresponds to going down and left while [3,2,0,1] corresponds to going up a little bit.

We qualitatively analyze LAPA's coarse-grained planning through a closed-loop rollout using a pre-trained model without action finetuning. Since latent actions aren't directly executable, we condition the current observation x_1 and LAPA's predicted latent action with the decoder of the quantization model. As shown in Figure 10 in Appendix, when instructed to ”take the broccoli out of the pot,” LAPA generates robot trajectories that reach for the broccoli, grab it, and, as the arm moves away, the broccoli disappears. This demonstrates LAPA's potential as a general-purpose robotic *world model*, predicting both actions and their outcomes.

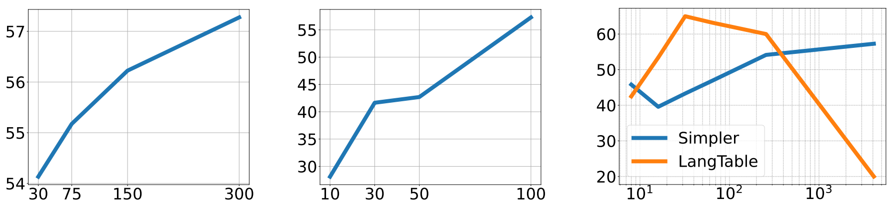
D Related Work

Vision-Language-Action Models Vision-Language Models (VLMs), trained on large-scale internet datasets have shown strong capabilities in understanding and generating both text and multimodal data [18, 19, 20, 21]. Leveraging this, recent advancements have introduced Vision-Language-Action Models (VLAs), which extend VLMs by fine-tuning them with robotic action data [1, 2, 22, 3]. Incorporating auxiliary objectives, such as visual traces [23], language reasoning paths [4], or creating conversational-style instruction datasets [24], have further improved VLA performance. However, these methods remain dependent on labeled action data. In contrast, our approach reduces reliance on human-teleoperated data by requiring labeled actions only for fine-tuning.

Training Robot Policies From Videos Videos offer rich data for robot learning, but most lack action labels [6]. Related work pretrains a vision encoder on egocentric human videos [25, 26, 27], or video generative models to generate future robot trajectories [28, 29]. Methods also extract diverse features from human videos such as interactions [30], affordances [31, 32, 33, 34], or visual traces [35, 36]. Some perform retargeting of human motions to robot actions [37, 38, 34, 39, 40, 41] or motion capture systems [42]. Finally, some train inverse dynamics models (IDMs), optical flow, or reinforcement learning models that predict actions from future state rollouts generated by world models [43, 44, 45, 46, 47].

Latent Actions Previous works have employed latent actions across diverse scenarios. GENIE [9] maps user inputs (ground-truth actions) to a latent space, allowing generative models to create interactive environments. We adopt a similar latent action model but apply it to label actionless data for training a VLA to solve robotic tasks. Similarly, some works use latent actions to pretrain and fine-tune policies for video games [48, 49, 50]. In contrast, we focus on learning latent actions from real-world human motions for more complex, continuous robotic tasks. Unlike other work that leverages latent actions by converting ground-truth actions into latent actions [51, 52, 53, 54], our approach derives latent actions directly from observations.

E Scaling Model, Data, and Latent Action Size



(a) Model Scaling (b) Data Scaling (%) (c) Latent Action Space

Figure 5: **Scaling Ablation Results of LAPA**. We scale 3 dimensions of LAPA: model parameters (in millions), data size (ratio among Bridgev2), and the latent action representation space, and show the downstream average success rate (%) on the SIMPLER fine-tuning tasks.

Large Language Models (LLMs) have demonstrated scaling laws [55], where performance improves with increases in model size, dataset size, and computational resources used for training. Similarly, we attempt to analyze whether LAPA benefits from scaling across three dimensions: latent action quantization model size, data size, and latent action representation space. For a controlled setup, we apply our method to Bridgev2 and then fine-tune it on SIMPLER except for Language Table result of Figure 5c.

As shown in Figure 5, scaling benefits LAPA across the three dimensions. Interestingly, we observe that the optimal scale of the latent action space depends on the complexity of the action dimension contained in the pretraining dataset. For example, increasing the latent action size for Language Table pretraining eventually harms the performance after a certain point. Except for Language Table, we maintain the generation space of LAPA at 8^4 throughout all of our main experiments. These results imply that when scaling pretraining to Internet-scale videos that go beyond manipulation videos, scaling LAPA in terms of model, dataset, and latent action space could improve performance, especially to capture higher action dimensions such as whole-body locomotion and manipulation.

F Limitations

We still face certain limitations. First, LAPA underperforms compared to action pretraining when it comes to fine-grained motion generation tasks like grasping. We believe that increasing the latent action generation space could help address this issue. Second, similar to prior VLAs, LAPA also encounters latency challenges during real-time inference. Adopting a hierarchical architecture, where a smaller head predicts actions at a higher frequency, could potentially reduce latency and improve fine-grained motion generation. Lastly, while we qualitatively demonstrate that our latent action space captures camera movements (Figure 13), we have not yet explored the application of LAPA beyond manipulation videos, such as those from self-driving cars, navigation, or landscape scenes. We leave these explorations for future work. We hope that our work can help overcome the data bottleneck in robotics and accelerate the development of generalist robot policies.

G Latent Action Quatization Model Details

We show model architecture details of our latent action quantization model in Figure 6. We utilize the C-ViViT model architecture from Villegas et al. [56] to replicate the latent action model from GENIE [9]. After latent model training, we utilize the z_2 as the latent action label for x_1. The encoder can be seen as the inverse dynamics model and the decoder can be seen as the world model.

H Experimental Setup

We evaluate the effectiveness of LAPA on 9 different task categories in 2 different simulation environments and 3 different real-world robotic tasks. Table 2 shows an overview of the pretraining and

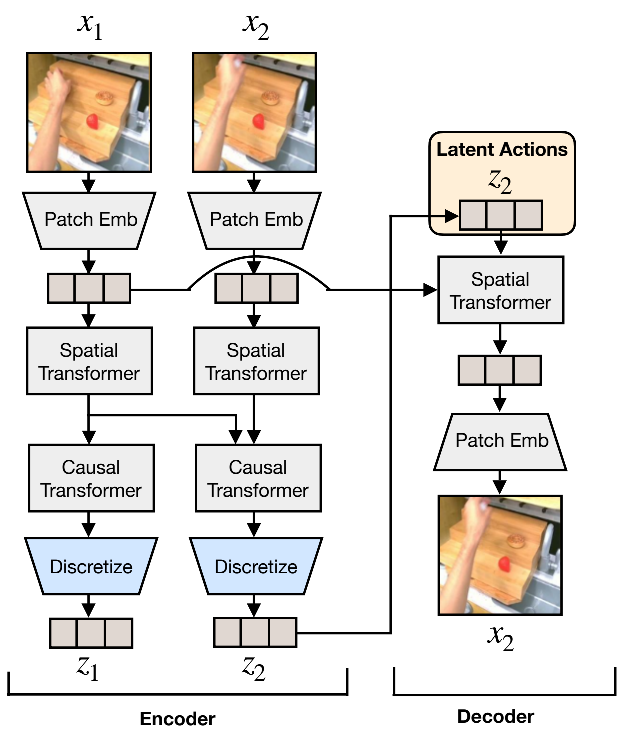


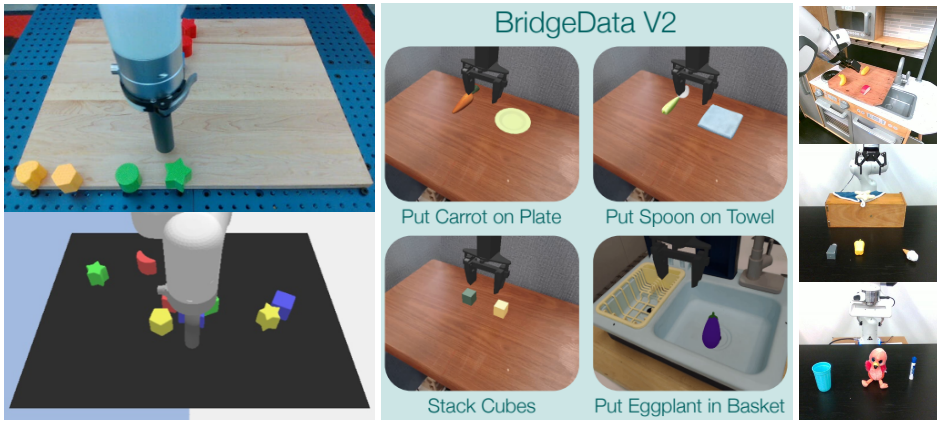
Figure 6: **Model architecture of our Latent Action Quantization Model.**

Table 2: **Pretraining and fine-tuning dataset for each environment.** Cross-Env denotes cross-environment, Cross-Emb denotes cross-embodiment, and Multi-Emb denotes multi-embodiment. For fine-tuning, MT denotes multi-task training and MI denotes tasks with diverse multi-instructions.

Environment	**Category**	**Pretraining**		**Fine-tuning**	
		Dataset	# Trajs	Dataset	# Trajs
LangTable	In-Domain	Sim (All 5 tasks)	181k	5 Tasks (MT, MI)	1k
	Cross-Task	Sim (All 5 tasks)	181k	1 Task (MI)	7k
	Cross-Env	Real (All 5 tasks)	442k	5 tasks (MT, MI)	1k
SIMPLER	In-Domain	Bridgev2	60k	4 Tasks (MT)	100
	Cross-Emb	Something v2	200k	4 Tasks (MT)	100
Real-World	Cross-Emb	Bridgev2	60k	3 tasks (MI)	450
	Multi-Emb	Open-X	970k	3 tasks (MI)	450
	Cross-Emb	Something v2	200k	3 tasks (MI)	450

fine-tuning dataset for each setup and Figure 7 visualizes the simulation benchmark and real-world setups.

Language Table [13] is a simulation where a robot performs 2 DOF actions to push blocks with 5 subtask categories (see Figure 7) (a)). Figure 7 (a) shows examples of the Language Table setup. During evaluation, we evaluate models for both *seen* and *unseen* scenarios, where *unseen* includes new objects (color and shape) and unseen combinations of seen objects. It includes 5 subtask categories: BlocktoBlock, BlocktoAbsolute, BlocktoBlockRelative, BlocktoRelative, and Separate. For Language Table experiments, we train VLA-based models to generate language directions (e.g. 'move up') before actual actions following Belkhale et al. [57], which significantly improved the



(a) LANGUAGE TABLE (b) SIMPLER (c) REAL

Figure 7: **Experimental Setups**. (a) shows an example from the 440k real-world trajectories (top) and the 181k simulation trajectories (bottom) from the Language Table Benchmark. (b) shows the 4 different evaluation tasks we use with the SIMPLER environment. (c) shows the three different tasks that we perform in the real-world.

performance [1]. For evaluation, we evaluate on 50 evaluation rollouts for each subtask category where the initial locations of the objects are randomized for each evaluation. Further details can be found in https://github.com/google-research/language-table.

SIMPLER [14] is a set of simulated environments for evaluating generalist robot manipulation policies. We assess our models on 4 tasks (Figure 7 (b)) using the 7 DOF WidowX robot arm. Since SIMPLER lacks fine-tuning trajectories, we collect 100 multi-task trajectories using successful rollouts from a VLA model trained on BridgeV2 data [17]. Figure 7 (B) shows examples of the SIMPLER setup. The SIMPLER environment does not provide any fine-tuning data for their evaluation pipeline, Thus, we first train our underlying VLM on the Bridgev2 dataset and perform zero-shot rollout on the 4 tasks in SIMPLER. Note that we use held-out trajectories differing in object orientation and position from the evaluation setup. We filter 25 successful trajectories for each task (total of 100) and use them as the fine-tuning dataset for all of our experiments. For evaluation, we evaluate on 24 rollouts per task while randomizing the initial object locations. We consider Bridgev2 and SIMPLER to be *in-domain* since they show a high correlation between real-world and simulation results with their simulation benchmark. Further details can be found in https://github.com/simpler-env/SimplerEnv.

Real-World Tabletop Manipulation experiments used a 7 DOF Franka Emika Panda robot arm in three environments (shown in Figure 7 (c)). We utilize three pretraining data sources: Bridgev2 [17], Open-X [3], and Something Something v2 [15]. Following Kim et al. [2], we finetune on three multi-instruction tasks: (1) 'Pick <object> into Sink', (2) 'Cover <object> with Towel', and (3) 'Knock <object> Over'. Each task involves 150 trajectories across 15 objects. We use a task-specific partial success criterion for evaluation, following Kim et al. [2]. Figure 7 (C) shows examples of the real-world tabletop manipulation experimental setup. For the teleoperation, we use the polymetis robotic stack[2] to collect 150 trajectories for each of the tasks. All of the tasks require multi-instruction following capabilities since there are 3 objects in the scene and the model has to condition on the task description to infer which object to interact with. Figure 8 shows samples of each task. For each task, we aim to quantify 3 distinct capabilities:

(1) We test the ability to infer the correct object from the task description between an unseen combination of seen objects during fine-tuninig, (2) We test the ability to infer the correct object from totally unseen objects during fine-tuning that may or may have not been observed during pretraining.

[1]For 7 DOF robot experiments, we found the benefit of generating language directions to be marginal compared to the increased inference cost. Therefore, we only generate delta end-effector actions on other experiments.

[2]https://github.com/facebookresearch/polymetis

Specifically, the *knocking* tasks was conducted with real-world objects that were highly unlikely to have been in any of the pertaining datasets. (3) We test the ability to infer the correct object (among seen objects, unseen combinations) from a totally unseen instruction that requires semantic reasoning (e.g. Pick up a spicy object). For each evaluation criteria, 6 rollouts are performed for each models, resulting in a total of 18 rollouts for each task category. Since there are three tasks, each model is evaluated with 54 rollouts in the real-world. We provide the full list of all of the seen and unseen objects used for each rollout in Table 13, 14, 15, and the total average success rates in Table 16.

Furthermore, for a fair comparison, we match the image resolution during training of all of our models and use the exact same object initial positions for all of our evaluation, mostly on the same day to minimize variability. For evaluation metrics, we adapt a partial success criteria for fine-grained evaluation, following Kim et al. [2], which we describe in detail below.

Knock down the <object>.

For knocking, we give 0.5 partial score if the robot reaches to the correct object and 1 if the robot knocks down the correct object.

Cover the <object> with a towel.

For covering, we give 0.33 partial score if the robot picks up the towel correctly, 0.66 if the robot reaches to the correct object or if the towel partially covers the object, and 1 if the correct object is completely covered by the towel.

Pick up the <object> and put it in the sink.

For pick and place, we give 0.25 for reaching to the correct object, 0.5 for grasping the object, 0.75 for grasping and moving the object towards the sink, but failing to place the object in the sink, and 1 for placing the correct object in the sink.

I Baseline Models

For the underlying VLM, we use the 7B Large World Model (LWM-Chat-1M) [20].

SCRATCH denotes the baseline model where we finetune our backbone VLM only on the downstream tasks, to quantify the gains we get from the pretraining stage.

UNIPI [43] uses a video diffusion model during pretraining to generate video rollouts given a language instruction, which does not require any action labels during pretraining similar to our approach. For finetuning, an inverse dynamics model (IDM) is trained to extract the ground truth actions given adjacent frames. We also finetune the diffusion model on the downstream task to match the target distribution. We use diffusion model from Ko et al. [44] which can be trained on 4 A100 GPUs. For all experiments, we train with 128 batch. We use the same inverse dynamics model as VPT during inference. To mediate estimation errors between the predicted video plans and executed actions being accumulated, we periodically conduct replanning by regenerating new video plans after executing two actions.

VPT [47] trains an IDM on action labeled data, and uses the IDM model to extract pseudo actions on raw videos. Then, we use the pseudo actions labeled by the IDM to pretrain our backbone VLM on the pretraining data, identical to Latent Pretraining of LAPA. We use ResNet18 followed by an MLP layer for the inverse dynamics model(IDM). The IDM is trained to predict an action when given two frames on a single A6000 GPU using using Adam optimizer with a learning rate 1e-4.

ACTIONVLA denotes the baseline that uses the actual ground-truth robot action labels during pretraining with the same backbone VLM. **ACTIONVLA** denotes the baseline that uses the actual ground-truth robot action labels during pretraining with the same backbone VLM. For ACTIONVLA and LAPA, we train with a batch size of 128 and with image augmentation for real-world finetuning. This may be seen as the upper bound, since it utilizes the actual ground-truth labels.

OpenVLA [2] is a state-of-the-art VLA model that was pretrained on 970k real-world robot demonstrations from the Open X-Embodiment Dataset and having a comparable model size to LAPA (7B). We compare against OpenVLA for real-world robot experiments by fine-tuning the pretrained OpenVLA on our downstream tasks. For OpenVLA (Bridge), we pretrain on Bridgev2 for 30 epochs with a batch size of 1024. For OpenVLA (Open-X), we use the pretrained checkpoint from Kim et al. [2]. For finetuning, we use LoRA finetuning [58] with batch size of 32. We have observed that full-finetuning and lora finetuning leads to similar performance, so we use LoRA finetuning as default for efficient fine-tuning. We finetune the model until the training action accuracy reaches 95%.

J Experimental Result Analysis

Table 3: **Pretraining trajectories statistics for downstream tasks.** Number of trajectories that are the same task with evaluation task for each pretraining dataset: Bridgev2, Open-X, and Something Something V2 (Sthv2) dataset.

Task	Bridgev2	Open-X	Sthv2
Knocking	2	7,969	6,655
Covering	898	5,026	6,824
Pick & Place	10,892	911,166	3,272

We further analyze the real-world robot results shown in Figures 2, focusing on how the task distribution in pretraining data impacts downstream performance. Table 3 presents the number of trajectories corresponding to each evaluation task (Knocking, Covering, and Pick & Place) across pretraining datasets (Bridgev2, Open-X, and Something Something V2 (Sthv2)), determined through *lexical* matching. We expect future work to use other methods of analyzing the relationship between pertaining and fine-tuning task distributions that capture *semantics* of the task rather than simple lexical matching. We perform this analysis to get a sense of how the task distribution in the pretraining data affects downstream task performance.

Knocking There are almost no knocking-related trajectories in Bridgev2. This scarcity may explain why models trained on Bridgev2 performed worse compared to those trained on Sthv2, despite a larger embodiment gap in the Sthv2 dataset (Figure 2).

Covering A similar trend is observed for the covering task. Given that the number of covering trajectories in Bridgev2 is relatively small compared to the Sthv2 dataset, models trained on Bridgev2 occasionally underperform compared to LAPA trained on Sthv2.

Pick & Place For the pick and place task, the trend reverses. The number of pick and place tasks in Sthv2 is relatively small compared to Bridgev2 and Open-X, which might explain why LAPA trained on Sthv2 significantly underperforms models trained on Bridgev2 or Open-X. Based on these results, we expect that pretraining on videos encompassing a wide range of skills will lead to a more robust generalist policy compared to training on robot videos with narrower skill sets. We also expect future research to provide a more in-depth analysis of the relationship between task distribution in pretraining data and performance on downstream tasks.

We also present the win rate of LAPA (Open-X) against OpenVLA (Open-X). As illustrated in Figure 9, LAPA outperforms OpenVLA in 65.4% when disregarding the ties. When considering the ties, LAPA outperforms OpenVLA in 31.5% of cases, while OpenVLA prevails in only 16.7%. Interestingly, they tie in 51.9% of the trials, suggesting that in about half the instances, both models either fail or achieve a similar partial success score. Note that these evaluations were performed while ensuring that the target and distractor objects were in identical initial locations during evaluation, alternating the models during evaluation. These results provide insight into the statistical significance of the comparison, supporting the use of multiple metrics to ensure a more comprehensive evaluation of physical robot performance in real-world scenarios [59], not only the average success-rate across all of the evaluation rollouts.

K Detailed Latent Action Analysis

We provide further qualitative analysis of LAPA. First, we analyze latent actions learned from Language Table with vocabulary size of 8 and sequence length of 1. In Figure 11, we show that each latent action corresponds to a semantic action (0: Move left and forward, 1: Move left and back, 2: Move right and back, 3: Move right slightly, 4: Move right, 5: Move back, 6: Do not move, 7: Move forward). We observe that increasing the latent action vocabulary size leads to capturing a more fine-grained information. We analyze the relationship between latent actions with ground-truth 2 DOF actions by mapping each instance into latent action space. As shown in Figure 12, we observe that latent actions are well-clustered in the actual 2D action space, indicating that latent actions are meaningful representations that are highly related to actual continuous actions.

We further analyze the latent actions learned from human manipulation videos using the Something-Something V2 dataset. As illustrated in Figure 13, these latent actions capture not only hand movements but also camera movements. Since the camera viewpoint varies throughout the videos in the Something-Something V2 dataset due to the videos being egocentric, our latent action quantization model also learns to represent camera movements. For instance, latent actions [3,5,2,7] and [5,6,7,6] correspond to slight downward camera movement, [4,0,0,4] and [2,3,6,6] indicate rightward movement, and [4,2,0,0] and [5,7,0,5] represent subtle upward camera shifts.

L Detailed Experimental Results

L.1 Language Table

We provide the detailed results of the experiments performed on the Language Table benchmark in Table 4, 5, 6, 7, 8, 9. For all of the tables in the appendix, we **bold** the best result among the comparisons and underline the second best. Each value denotes the success rate (%). 50 evaluation rollouts are performed for each task category, resulting in 250 total evaluation rollouts per model for each table.

We also show the qualitative result of UNIPI where the diffusion model generates the correct plan for simple and short-horizon tasks (e.g. separate tasks). However, the diffusion model generates the wrong plan corresponding to the instruction when the task requires longer horizon planning (Figure 14).

Table 4: **Language Table In-Domain Seen Results.**

	SCRATCH	UNIPI	VPT	LAPA	ACTIONVLA
Block2Block	4.0	14.0	36.0	<u>58.0</u>	**76.0**
Block2Absolute	6.0	4.0	38.0	<u>56.0</u>	**72.0**
Block2BlockRelative	10.0	12.0	<u>48.0</u>	52.0	**76.0**
Block2Relative	6.0	10.0	26.0	<u>48.0</u>	**70.0**
Separate	52.0	72.0	70.0	**96.0**	<u>90.0</u>
AVG	15.6	22.4	43.6	<u>62.0</u>	**76.8**

L.2 SIMPLER

We provide detailed results of various models evaluated on SIMPLER environment. Table 10 shows the setting where baseline models are pretrained on Bridgev2 and then finetuned on SIMPLER rollouts (100 videos). The results show detailed results for each task (stack green to yellow block, put carrot on plate, put spoon on otowel, put eggplant in basket) and subtasks (grasping and moving).

We also provide detailed results of the setting where baseline models are pretrained on human manipulation videos (Something Something V2 dataset) and then finetuned on SIMPLER rollouts (100

Table 5: **Language Table In-Domain Unseen Results.**

	SCRATCH	UNIPI	VPT	LAPA	ACTIONVLA
Block2Block	8.0	4.0	26.0	50.0	**62.0**
Block2Absolute	10.0	6.0	42.0	48.0	**58.0**
Block2BlockRelative	2.0	6.0	20.0	28.0	**48.0**
Block2Relative	8.0	6.0	32.0	38.0	**44.0**
Separate	48.0	44.0	44.0	**84.0**	82.0
AVG	15.2	13.2	32.8	49.6	**58.8**

Table 6: **Language Table Cross-Task Seen Results.**

	SCRATCH	UNIPI	VPT	LAPA	ACTIONVLA
Block2Block	18.0	12.0	74.0	74.0	**76.0**
Block2Absolute	8.0	6.0	56.0	62.0	**72.0**
Block2BlockRelative	6.0	2.0	62.0	72.0	**76.0**
Block2Relative	24.0	16.0	**72.0**	60.0	70.0
Separate	80.0	68.0	96.0	**98.0**	90.0
AVG	27.2	20.8	72.0	73.2	**76.8**

Table 7: **Language Table Cross-Task Unseen Results.**

	SCRATCH	UNIPI	VPT	LAPA	ACTIONVLA
Block2Block	16.0	4.0	**66.0**	46.0	62.0
Block2Absolute	10.0	10.0	56.0	52.0	**58.0**
Block2BlockRelative	8.0	10.0	46.0	**48.0**	**48.0**
Block2Relative	12.0	4.0	**52.0**	38.0	44.0
Separate	66.0	52.0	84.0	**90.0**	82.0
AVG	22.4	16.0	**60.8**	54.8	58.8

Table 8: **Language Table Cross-Environment Seen Results.**

	SCRATCH	UNIPI	VPT	LAPA	ACTIONVLA
Block2Block	4.0	4.0	16.0	26.0	**66.0**
Block2Absolute	6.0	4.0	8.0	16.0	**58.0**
Block2BlockRelative	10.0	8.0	6.0	20.0	**62.0**
Block2Relative	6.0	4.0	12.0	22.0	**54.0**
Separate	52.0	48.0	48.0	**84.0**	**84.0**
AVG	15.6	13.6	18.0	33.6	**64.8**

Table 9: **Language Table Cross-Environment Unseen Results.**

	SCRATCH	UNIPI	VPT	LAPA	ACTIONVLA
Block2Block	8.0	2.0	2.0	30.0	**38.0**
Block2Absolute	10.0	6.0	4.0	14.0	**48.0**
Block2BlockRelative	2.0	6.0	2.0	10.0	**50.0**
Block2Relative	8.0	4.0	40.0	18.0	**54.0**
Separate	48.0	42.0	44.0	76.0	**80.0**
AVG	15.2	12.0	18.4	29.6	**54.0**

videos) in Table 11. We only compare to UNIPI, VPT, and LAPA since ACTIONVLA could not be trained without ground-truth action labels.

L.3 Real-world

We provide the detailed result of real world evaluation depending on the generalization type: (1) seen objects but unseen combinations, (2) unseen objects, and (3) seen objects but unseen instructions. The results are shown in Table 12. As shown in the table, LAPA (Open-X) outperforms OpenVLA (Open-X) on all types of generalization settings. Also, LAPA (Human Videos) shows

Table 10: **SIMPLER results of Bridgev2 Pretraining.** Success, Grasping, and Moving Rates (%) in SIMPLER environment. We pretrain UNIPI, VPT, and LAPA on Bridgev2 dataset without using ground-truth action labels and ACTIONVLA on Bridgev2 using action labels. The main 4 tasks are: stack green to yellow block, put carrot on plate, put spoon on towel, and put eggplant in basket. Best is **bolded** and second best is underlined.

Success Rate	SCRATCH	UNIPI	VPT	LAPA	ACTIONVLA
Stack G2Y	29.2	2.7	45.8	<u>54.2</u>	**75.0**
Carrot2Towel	29.2	2.7	37.5	<u>45.8</u>	**58.0**
Spoon2Plate	50.0	0.0	**70.8**	**70.8**	**70.8**
Eggplant2Bask	29.2	0.0	<u>50.0</u>	**58.3**	50.0
AVG	34.4	1.3	51.0	<u>57.3</u>	**63.5**
Grasping Rate					
Grasp Green Block	<u>66.6</u>	20.8	62.5	62.5	**87.5**
Grasp Carrot	45.8	33.2	<u>54.1</u>	58.3	**75.0**
Grasp Spoon	70.8	22.2	79.2	**83.3**	**83.3**
Grasp Eggplant	62.5	16.0	70.8	**83.3**	<u>75.0</u>
AVG	61.4	23.1	<u>66.7</u>	71.9	**80.2**
Moving Rate					
Move Green Block	58.3	29.1	58.3	<u>66.6</u>	**91.6**
Move Carrot	45.8	48.6	<u>66.6</u>	70.8	**91.6**
Move Spoon	70.8	34.6	79.2	**83.3**	<u>79.2</u>
Move Eggplant	<u>87.5</u>	58.0	70.8	<u>87.5</u>	**91.6**
AVG	65.6	42.6	68.7	<u>77.1</u>	**88.5**

Table 11: **SIMPLER results of Human Manipulation Video Pretraining.** Success, Grasping, and Moving Rates (%) in SIMPLER environment. We pretrain UNIPI, VPT, and LAPA on Something-Something V2 dataset without using ground-truth action labels. The main 4 tasks are: stack green to yellow block, put carrot on plate, put spoon on towel, and put eggplant in basket. Best is **bolded** and second best is underlined.

Success Rate	VPT	UNIPI	LAPA
StackG2Y	**50.0**	0.0	**50.0**
Carrot2Towel	<u>29.1</u>	1.3	**50.0**
Spoon2Plate	<u>37.5</u>	1.3	**50.0**
Eggplant2Bask	**66.6**	0.0	<u>58.3</u>
AVG	<u>45.8</u>	0.7	**52.1**
Grasping Rate			
Grasp Green Block	**66.6**	2.7	<u>58.3</u>
Grasp Carrot	<u>45.8</u>	31.7	**62.5**
Grasp Spoon	<u>70.8</u>	21.7	**75.0**
Grasp Eggplant	**91.6**	6.8	<u>70.8</u>
AVG	**68.7**	15.7	<u>66.7</u>
Moving Rate			
Move Green Block	**62.5**	2.7	**62.5**
Move Carrot	<u>58.3</u>	37.5	**70.8**
Move Spoon	<u>54.1</u>	18.1	**75.0**
Move Eggplant	**91.6**	<u>50.3</u>	83.3
AVG	<u>66.6</u>	27.1	**72.9**

good generalization performance, especially for unseen objects. We conjecture that this is because Something Something V2 dataset interacts with much diverse objects compared to Bridgev2.

We also provide the full list of objects and the partial success recorded for each of the evaluation rollout: Knocking (Table 13), Covering (Table 14), and Pick & Place (Table 15). The total average success rate is provided in Table 16).

Table 12: **Evaluation Results divided into eval types**. We average the success rate across the 3 tasks depending on what capability we are trying to quantify: (1) seen objects but unseen combinations, (2) unseen objects, and new instructions requiring semantic reasoning. Best is **bolded** and second best is underlined.

	Seen Obj. Unseen Combo	Unseen Obj.	Seen Obj. Unseen Instr.	AVG
SCRATCH	18.0	20.3	25.4	21.2
ACTIONVLA (Bridge)	38.3	31.8	27.7	32.6
OPENVLA (Bridge)	35.6	34.6	22.1	30.8
LAPA (Bridge)	43.4	31.4	35.6	36.8
OPENVLA (Open-X)	46.2	42.1	43.4	43.9
LAPA (Open-X)	**57.8**	**43.9**	**48.5**	**50.1**
LAPA (Human Videos)	36.5	37.4	28.1	34.0

Table 13: **Knocking Task Results**

	OpenVLA (OpenX)	LAPA (OpenX)	OpenVLA (Bridge)	ActionVLA (Bridge)	LAPA (Bridge)	Scratch	LAPA (Sthv2)
Seen							
flamingo	0	0.5	0.5	0.5	0	0	0.5
pistachios	0.5	1	0.5	0	1	0	1
soft scrub	0	0	0	0	0.5	0	0.5
white cup	1	0	0	0.5	0.5	0.5	0
mustard	0	1	0	0	0	0	0
water bottle	1	1	0.5	0	0	0.5	0
SUM	2.5	3.5	1.5	1	2	1	2
Unseen							
pringles	0.5	0.5	0.5	0	0	0	0
hersey's chocolate syrup	0	0	0	0	0	0	0
popcorn	0	1	1	1	1	0	1
skittles	0	0	0	0	0	0	0
green board marker	0.5	0.5	0.5	0.5	0.5	0.5	0.5
paper towel	0	0	0	0	0	0	0
SUM	1	2	2	1.5	1.5	0.5	1.5
Seen Semantic							
a drink that contains orange	0	0	0	0	0	0	0
food to eat with milk	0.5	0	0	0	0	0	0
a object used for cleaning	0	1	0	0	0	0	0
something to wash dishes	1	1	0	0.5	1	0.5	0
the nuts	1	1	0.5	1	1	0.5	1
rectangle object	1	1	0.5	0.5	0.5	0	1
SUM	3.5	4	1	2	2.5	1	2
Success Rate (Strict)	27.78%	**44.44%**	5.56%	11.11%	22.22%	0.00%	22.22%
Success Rate	38.89%	**52.78%**	25.00%	25.00%	33.33%	13.89%	30.56%
Reaching Success Rate	50.00%	**61.11%**	44.44%	38.89%	44.44%	27.78%	38.89%

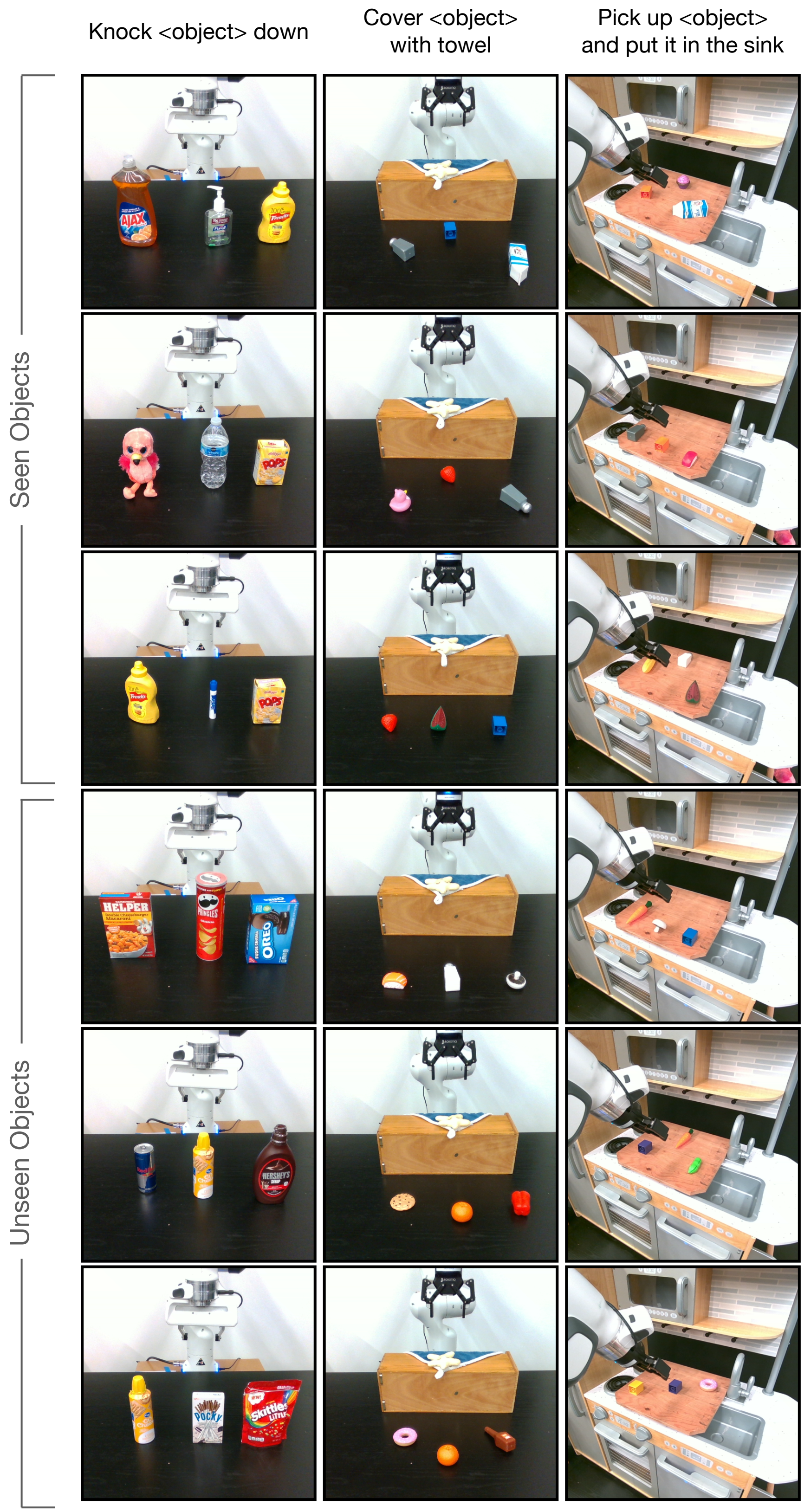
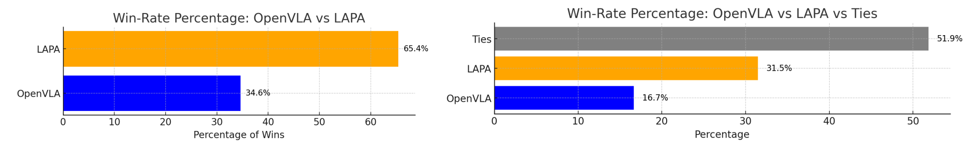


Figure 8: **Real-world Tabletop Manipulation Examples.**



(a) **Win rate (%) disregarding ties.** (b) **Win rate (%) with ties.**

Figure 9: **Pairwise win rate (%)**. We compare a pairwise win-rate of OpenVLA and LAPA across the 54 evaluation rollouts in the real-world. (a) shows the win-rate while ignoring the ties and (b) shows the ties together with the individual wins.

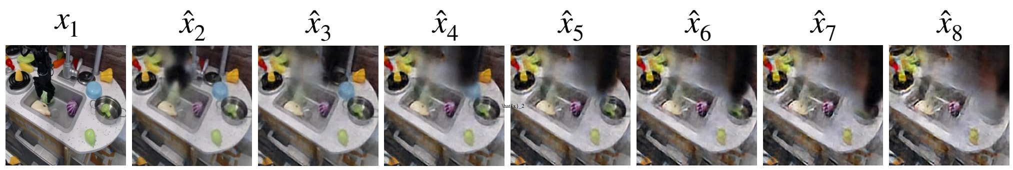


Figure 10: **Closed loop rollout of LAPA.** LAPA is conditioned on current image x_1 and language instruction of 'take the broccoli out of the pot'. We generate rollout images by conditioning the decoder of Latent Action Quantization Model with latent actions generated by LAPA.

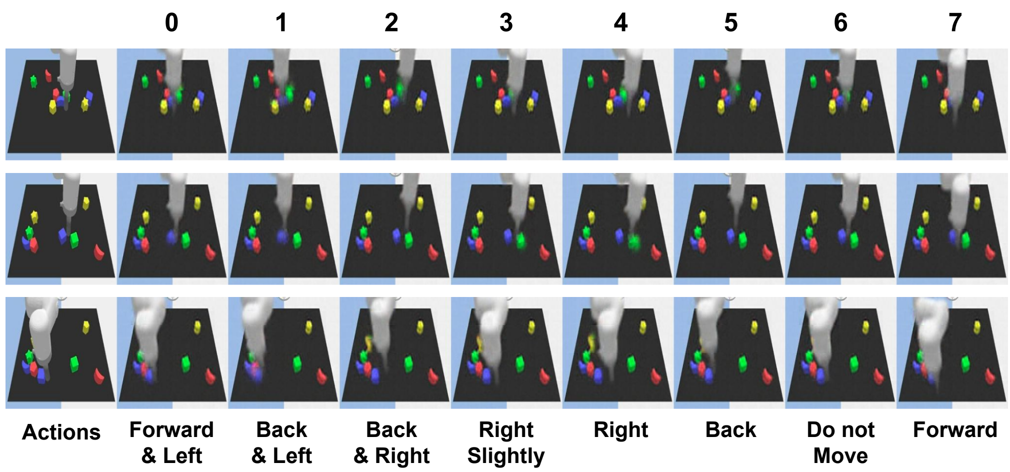


Figure 11: **Latent Action Analysis in Language Table.** We condition the current observation x_1 and quantized latent action to the decoder of the latent action quantization model. We observe that each latent action can be mapped into a semantic action. For example, latent action 0 corresponds to moving a bit left and forward and corresponds to moving a bit left and back.

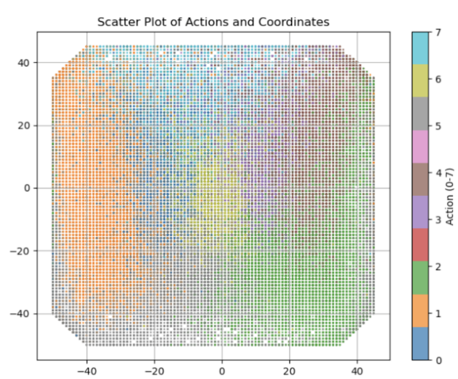


Figure 12: **Correlation of latent action with ground-truth actions** When we map latent actions to ground-truth 2 DOF actions of Language Table, we observe that latent actions are well-clustered in the actual 2D action space.

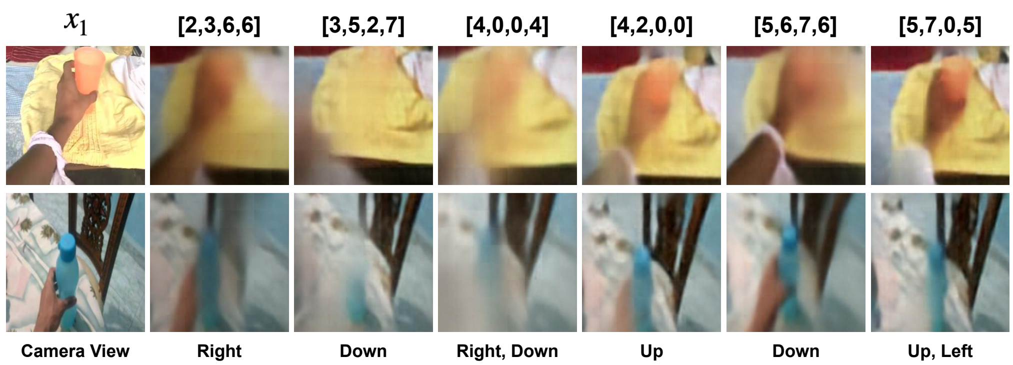


Figure 13: **Latent Action Analysis in Human Manipulation Videos.** We condition the current observation x_1 and quantized latent action to the decoder of the latent action quantization model. We observe that each latent action can be mapped into a semantic action including camera movements. For example, latent action [3,5,2,7] corresponds to moving the camera a bit down while [4,2,0,0] corresponds to moving the camera slightly up.

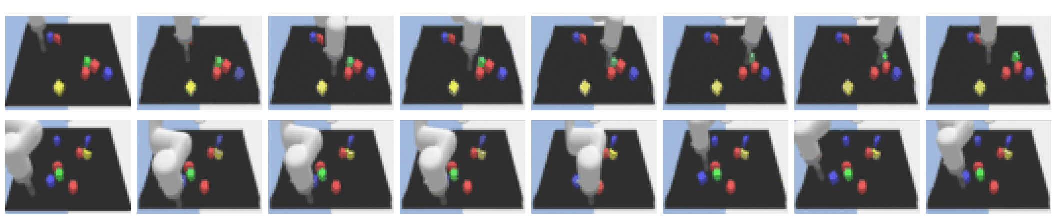


Figure 14: **Success and Failure Cases of UniPi.** (Top) Given the instruction of 'move the green block away from the red cube and red pentagon', the diffusion model of UniPi successfully generates the plan. (Bottom) Given the instruction of 'put the blue moon toward the yellow block', the diffusion model fails to generate the correct plan.

Table 14: **Covering Task Results**

	OpenVLA (OpenX)	LAPA (OpenX)	OpenVLA (Bridge)	ActionVLA (Bridge)	LAPA (Bridge)	Scratch	LAPA (Sthv2)
				Seen			
icecream	0.33	0.33	0.33	0.33	0.33	0.33	0
strawberry	0.33	1	0.33	1	0.33	1	1
pepper	0.33	0	0.33	0.33	0.33	0.33	0.33
watermelon	0.33	0.33	0.33	0.33	0.33	0	0.33
blue lego block	0.66	1	1	1	1	0.33	0.33
pink duck	0.33	1	0.33	0.33	0.33	0	0.33
SUM	2.31	3.66	2.65	3.32	2.65	1.99	2.32
				Unseen			
donut	0.33	1	0.66	1	0.66	0.66	0.33
orange	0.33	0.33	1	0	0.33	1	1
mushroom	0.33	0.33	0.33	0.33	0.33	0.33	0.33
yellow lego block	0.33	1	1	0.33	0	0.33	0.33
peas	1	0	0.66	1	1	0.33	1
egg	0	1	0.33	0	0.66	0	1
SUM	2.32	3.66	3.98	2.66	2.98	2.65	3.99
				Seen Semantic			
drink	0.33	0	0.66	1	0.33	0.33	0.66
yellow object	0.66	0.66	0	0	0.33	0	0.33
fruit	0.33	0.33	0.33	0.33	0.33	0.33	0.33
vegetable	0.33	0.33	0	0.33	0.33	0.33	0.33
edible object	0.33	0.33	0.66	0	0.33	1	0.33
condiment	0.33	0.33	0.33	0	0.33	0.33	0.33
SUM	2.31	1.98	1.98	1.66	1.98	2.32	2.31
Success Rate (Strict)	5.56%	**33.33%**	16.67%	27.78%	11.11%	16.67%	22.22%
Success Rate	38.56%	**51.67%**	47.83%	42.44%	42.28%	38.67%	47.89%
Reaching Success Rate	16.66%	**38.89%**	**38.89%**	27.78%	22.22%	22.22%	27.78%

Table 15: **Pick & Place Sink Task Results**

	OpenVLA (OpenX)	LAPA (OpenX)	OpenVLA (Bridge)	ActionVLA (Bridge)	LAPA (Bridge)	Scratch	LAPA (Sthv2)
Seen							
milk	1	1	1	1	1	0	1
orange lego block	1	1	0	1	0	0	0
ketchup	0.25	0.25	0.25	0.25	0	0	0
corn	1	0.75	1	0.25	0.25	0.25	0.25
icecream	0.25	0	0	0	1	0	1
salt	0	0.25	0	1	0	0	0
SUM	3.5	3.25	2.25	3.5	2.25	0.25	2.25
Unseen							
carrot	1	0.25	0	0.25	1	0.25	0.25
yellow paprika	1	1	0	0.25	0.25	0	1
yellow cube	1	0.5	0.25	0.5	0	0	0
salmon sushi	0	0.25	0	0.5	0	0	0
orange	1	0	0	0	0	0.25	0
blue cube	0.25	0.25	0	0	0	0	0
SUM	4.25	2.25	0.25	1.5	1.25	0.5	1.25
Seen Semantic							
an object that is yellow	1	1	0	1	0.25	0	0
an object that is round	0	0.25	0	0	0	0.25	0
an object that is a fruit	1	1	1	1	0	1	0.75
an object that you can drink	0	0.25	0	0.5	0	0	0
an object that is a vegetable	0	0	0	0	0	0	0
an object that is an animal	0	0.25	0	0.25	0.25	0	0
SUM	2	2.75	1	2.75	0.5	1.25	0.75
Success Rate (Strict)	**50.00%**	27.78%	16.67%	27.78%	16.67%	5.56%	16.67%
Success Rate	**54.17%**	45.83%	19.44%	43.06%	22.22%	11.11%	23.61%
Reaching Success Rate	66.67%	**83.33%**	27.78%	72.22%	38.89%	27.78%	33.33%

Table 16: **Summary of Total Success Rates (%)**

	OpenVLA (OpenX)	LAPA (OpenX)	OpenVLA (Bridge)	ActionVLA (Bridge)	LAPA (Bridge)	Scratch	LAPA (Sthv2)
Total Success Rate	43.87%	**50.09%**	30.76%	36.83%	32.61%	21.22%	34.02%
Total Success Rate (Strict)	27.78%	**35.19%**	12.96%	22.22%	16.67%	7.41%	20.37%