DemoSpeedup: Accelerating Visuomotor Policies via Entropy-Guided Demonstration Acceleration

Lingxiao Guo[1], **Zhengrong Xue**[2,1,3], **Zijing Xu**[4], **Huazhe Xu**[2,1,3]
[1] Shanghai Qi Zhi Institute, [2] Tsinghua University IIIS, [3]Shanghai AI Lab,
[4]University of Electronic Science and Technology of China
guolingxiao3@gmail.com, huazhe_xu@mail.tsinghua.edu.cn

Abstract: Imitation learning has shown great promise in robotic manipulation, but the policy's execution is often unsatisfactorily slow due to commonly tardy demonstrations collected by human operators. In this work, we present *DemoSpeedup*, a self-supervised method to accelerate visuomotor policy execution via entropy-guided demonstration acceleration. *DemoSpeedup* starts from training an arbitrary generative policy (e.g., ACT or Diffusion Policy) on normal-speed demonstrations, which serves as a per-frame action entropy estimator. The key insight is that frames with lower action entropy estimates call for more consistent policy behaviors, which often indicate the demands for higher-precision operations. In contrast, frames with higher entropy estimates correspond to more casual sections, and therefore can be more safely accelerated. Thus, we segment the original demonstrations according to the estimated entropy, and accelerate them by down-sampling at rates that increase with the entropy values. Trained with the speedup demonstrations, the resulting policies execute up to 3 times faster while maintaining the task completion performance. Interestingly, these policies could even achieve higher success rates than those trained with normal-speed demonstrations, due to the benefits of reduced decision-making horizons. Project Page: https://demospeedup.github.io/.

Keywords: Imitation Learning, Manipulation, Demonstration Acceleration

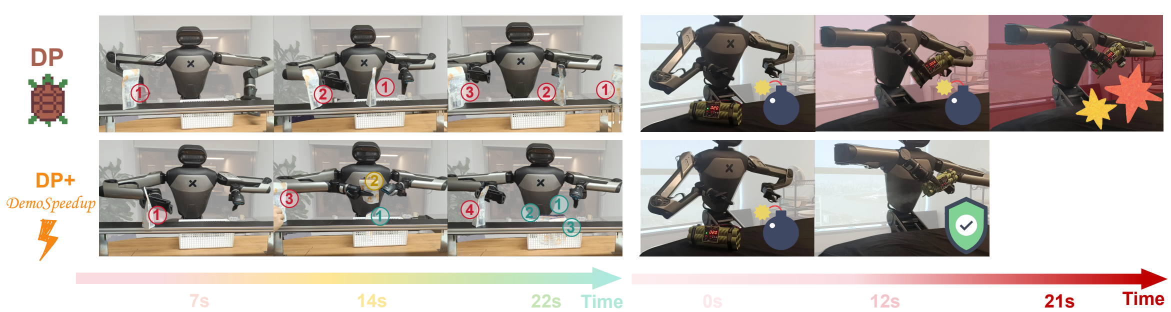

Figure 1: Manipulation speed is crucial for improving the productivity and ensuring the success of time-sensitive tasks. ***DemoSpeedup*** enables boosting the execution speed of visuomotor policy from slow demonstrations. Left: Original policy fails to track the speed of the production line and scan the products, while *DemoSpeedup* succeeds. Right: *DemoSpeedup* succeed to depose the time bomb toy before the end of countdown but the original policy fails.

1 Introduction

Imitation learning has achieved remarkable success for robot manipulation tasks from a perspective of task completion rates [1, 2, 3, 4], but visuomotor policies are often less satisfactory in terms of time efficiency. For visually pleasing fluency, it has become a common practice in the policy learning

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

community that the presented video demos are accelerated by $2\times \sim 10\times$ [1, 2, 5]. However, low efficiency might be a concern for some time-sensitive settings in the real world, e.g., caring for babies and the elderly or manufacturing on an assembly line.

Recently, some test-time policy acceleration techniques have been developed to improve execution speed by naively down-sampling the action chunk to be executed at test time [6, 7]. However, test-time acceleration often leads to a notable performance drop when the acceleration rate is relatively high (e.g., by $2\times$) due to the apparent distributional shift induced by speedup during deployment.

In this work, we assume that tardy demonstrations collected by human operators are the primary cause for the slow execution of behavior cloning policies. Empirically, we observed that even skillful data collectors with over 100 hours of experience struggle to teleoperate the robot arms as fluently as using their own hands. In both VR [8, 9, 10] and kinematics-teaching [11, 12, 13, 14] teleoperation, the lack of full-directional, non-occluded view of the 3D scene, as well as the absence of tactile proprioception, are the major obstacles to achieving faster motions. Besides, morphological heterogeneity between humans and robots, combined with equipment latency, further increases the difficulty of teleoperation and slows down the data collection speed.

In response to the tardiness of human-collected demonstrations, we introduce *DemoSpeedup*, designed to boost policies' execution efficiency without sacrificing their task completion performance. Rather than naively downsample the demonstrated trajectories by a constant rate, *DemoSpeedup* preserves the high-precision sections (e.g., picking up objects) and only accelerates in the low-precision sections (e.g., approaching objects in free air) to promote the task completion rate after acceleration.

The core of *DemoSpeedup* is an entropy-guided precision measurement mechanism, which allows the adaptive adjustment of the acceleration rate while avoiding the need for additional human annotations or hand-designed, task-specific priors. Our key observation is that human operators tend to have multiple casual yet reasonable choices in low-precision sections, and follow more consistent behaviors in high-precision sections to ensure successful manipulation. Therefore, *action entropy*, which reflects the randomness of the actions, could be an implicit indicator of the precision required. However, the discretized actions recorded in the demonstration trajectory are very sparsely located in the multi-dimensional continuous action space, which is the major obstacle to calculating the action entropy offline from the human-collected demonstration dataset.

In *DemoSpeedup*, we propose to overcome the obstacle by estimating action entropy from a self-supervised proxy policy, which can be an arbitrary generative behavior learning policy trained on the non-accelerated source dataset. The proxy policy is not responsible for action prediction, but is only used to help distill the action entropy information embedded in the source dataset. A clustering-based scheme is designed to process the proxy-inferred per-frame entropy into trajectory segmentation with precision labeling, ready for piecewise varying-speed acceleration. Finally, the accelerated dataset facilitates the training of an accelerated behavior cloning policy, which is the end product of the *DemoSpeedup* pipeline used for action prediction during deployment.

Empirically, we validate the effectiveness of *DemoSpeedup* by instantiating it with two popular generative behavior cloning policies: Action Chunking with Transformers (ACT) [15] and Diffusion Policy (DP) [1]. We conduct extensive experiments on a diverse range of tasks in the simulation and the real world. The results demonstrate *DemoSpeedup* strikes a $1.7\times \sim 3\times$ speedup in time efficiency, while obtaining task completion success rates on par with or sometimes even higher than the same policy trained on the non-accelerated dataset.

2 Related Work

Learning from human demonstrations. Learning from human demonstrations has emerged as a widely adopted approach in robotic manipulation. Generative policies trained by imitation learning, such as ACT [15] and Diffusion Policy [1] can strike a performance that matches the demonstrations. The generalization of imitation learning [16, 17, 18] has been boosted over the years. However, execution speed of current imitation learning paradigms is still restricted by the slow demonstrations.

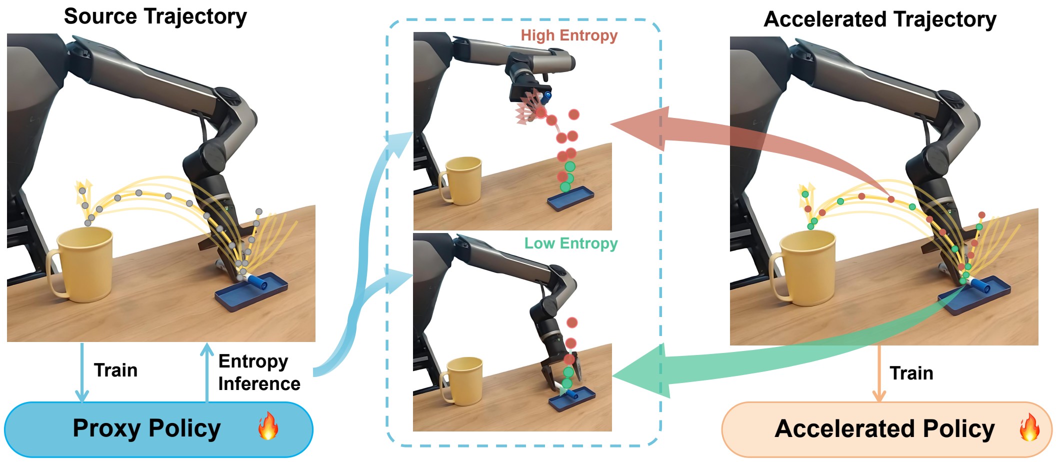

Figure 2: ***DemoSpeedup*** utilizes a generative policy trained from original demonstrations to estimate conditional action entropy. Actions with high entropy (red points) are down-sampled at a higher rate while actions with low entropy (green points) are down-sampled at a lower rate.

The problem still exists in large datasets [19, 20, 21, 22] collected by teleoperation. While VLAs trained with those large datasets [19, 23, 24, 25] exhibits strong generalization, numerous slow demonstrations within the data cause the learned policies to be much slower than normal human speed. This hinders the real-world application of robots in daily life.

Data curation for manipulation. Several prior works curate data aiming to improve the success rate of the trained policy. [26] train and cross-validate a classifier to discern successful policy rollouts from unsuccessful ones and use the classifier to filter heterogeneous demonstration datasets. [27, 28] down-sample the demonstrations with either geometric or human-labeled metrics, showing that by reducing episode length, the compounding error could drop [29] and the success rate can increase. But they relies on close-loop control and make the manipulation time longer. Overall, all these methods focus on improving success rates. In contrast, the objective of curating data in this work is to boost manipulation speed. At the same time, the episode length reduces, resulting in a potential effect to improve the performance.

3 Method

In this section, we present *DemoSpeedup*, an entropy-guided approach to accelerate demonstrations to speed up policy execution. As shown in Figure 2, *DemoSpeedup* first utilizes a proxy policy trained on source data to estimate the action entropy in a nonparametric way. Then it leverages a density-based cluster method to segment trajectories into high-precision and low-precision sets, followed by piecewise down-sampling at rates according to the precision level. Finally, some training and deployment details is introduced to guarantee the performance of accelerated policies.

3.1 Action Entropy Estimation

We leverage action entropy to reflect the precision level required for an action in human demonstrations. To estimate the conditional action entropy of demonstration frames, *DemoSpeedup* starts from training a proxy policy on the source dataset of original-speed demonstrations. We represent the behavior cloning policy as $\pi_\theta(A_t|o_t)$, where $A_t = \{a_t[t], a_t[t+1], ..., a_t[t+K-1]\}$ is the action chunk [30] and K is the chunk length. For entropy calculation, we sample N action chunk samples $\{a_t^i[t], ..., a_t^i[t+K-1]\}_{i=1}^N$ conditioned on the current observation o_t. Then, we perform Gaussian kernel density estimation [31] to obtain the probability density distribution of the actions conditioned on the current observation:

$$\hat{p}(a_t|o_t) = \frac{1}{NKh} \sum_{j=t-K+1}^{t} \sum_{i=1}^{N} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(a_t - a_j^i[t])^2}{2h^2}\right) \tag{1}$$

where h is the bandwidth. Then we estimate the conditional action entropy at o_t by:

$$\hat{H}(a_t|o_t) = -\sum_{j=t-K+1}^{t}\sum_{i=1}^{N}\hat{p}(a_j^i[t]|o_t)\log\hat{p}(a_j^i[t]|o_t) \tag{2}$$

We perform the per-frame operation along all timesteps for all the trajectories in the dataset. We instantiate the proxy policy with either Action Chunking with Transformers (ACT) [15] or Diffusion Policy (DP) [1]. For ACT, action samples are obtained by sampling different latent variables in the CVAE prior distribution, i.e, $x \sim \mathcal{N}(0, 1)$. For DP, action samples are generated by sampling multiple noise sequences given the observation.

3.2 Entropy-Guided Demonstration Acceleration

The estimated entropy paves the way for subsequent steps to identify the precision level of different segments and accelerate them at different rates. We develop a cluster-based approach to determine the precision level and leverage entropy for demonstration speedup.

Entropy preprocessing. As the teleoperation data can be very noisy and have harmful impact on clustering, we first utilize Isolation Forest [32] to detect the abnormal entropy values in one trajectory, after which the outliers are substituted by the adjacent normal values. Then, the entropy of each frame $\hat{H}(a_t|o_t)$ is first concatenated with its time index t to preserve the temporal property. All these obtained entropy points in one episode are normalized for clustering preparation.

Clustering for precision labeling. We adopt a density-based clustering method to divide those entropy points into fine-grained and coarse-grained areas. Specifically, we adopt hierarchical density-based clustering (Hdbscan) [33] to cluster those entropy points. Those high-entropy points are labeled to outliers, while low-entropy areas are labeled to clusters. To further exclude clustering noise, we simply filter all the obtained clusters by preserving clusters in which the mean entropy values are lower than zero. All the time indices in the preserved clusters are labeled as set P, i.e., precision set; and the rest are identified as set C, i.e., casualness set.

Replicate-before-downsample strategy. After getting precision labels, now it's possible to speed up the temporal segments at different rates by down-sampling. However, naively down-sampling the whole trajectory $\{(o_t, a_t)\}_{t=1}^{T}$ will significantly reduce the visited state diversity in the dataset, causing a severe waste of the demonstrations and empirically leading to a serious performance drop. To avoid the potential information loss caused by acceleration, we develop a simple replicate-before-downsample strategy, which retains all the observation frames that appear in the source dataset. More specifically, at an acceleration rate of $N\times$, the target chunk is replicated into N copies. The i-th copy is down-sampled by $N\times$ with a starting offset of i frames. Instead of skipping the intermediate frames, our strategy essentially splits the chunk into N accelerated sub-chunks, thus retaining the same diversity of the visited states as in the source demonstrations.

Geometrical consistency. Since action chunking has crucial impact on imitation learning performance, it is necessary to determine the chunk length of the accelerated policy trained on speedup demonstrations. We opt to keep the geometrical distance traveled by an action chunk roughly the same as the original policy. This ensures that the accelerated policy only needs to fit much less action labels than the original policy for the same segments in the demonstrations, which benefits for converging and reducing compounding error.

Controller requirements. During data collection and deployment, acceleration requires high-precision controller of the robot. Apparently, if the controller is inaccurate, the control dynamics could differ a lot between the original speed and speedup actions, which also leads to a performance drop. During evaluation, we find some robot gripper controllers fail to track high speed and cause failures. We simply increase the gripper gain to solve this problem.

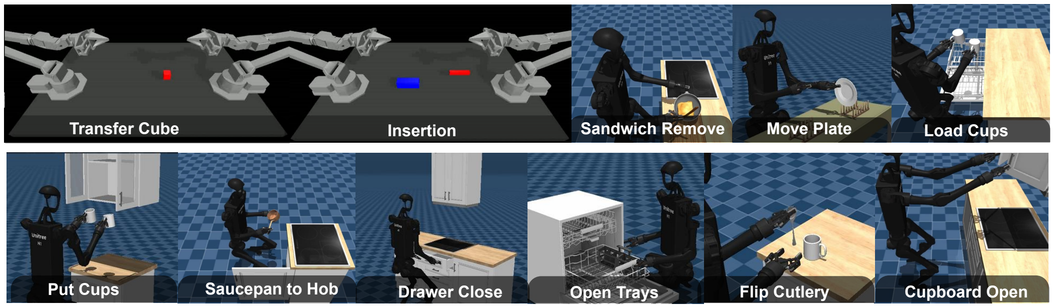

Figure 3: **Simulation tasks.** The environments are from Aloha and Bigym, featuring bimanual and mobile manipulation from human-collected datasets.

Method	**Transfer Cube**		**Insertion**		**Sandwich Remove**		**Move Plate**		**Load Cups**		**Put Cups**	
	success rate(↑)	episode len(↓)	success rate(↑)	episode len(↓)	success rate(↑)	episode len(↓)	success rate(↑)	episode len(↓)	success rate(↑)	episode len(↓)	success rate(↑)	success rate(↑)
ACT	72%	291	21%	452	53%	368	**54%**	157	**61%**	319	61%	288
ACT-2x	70%	162	13%	238	46%	193	46%	119	50%	195	54%	141
ACT+*DemoSpeedup*	**81%**	**121**	**30%**	**151**	**77%**	**156**	53%	**91**	59%	**176**	**62%**	**132**
DP	66%	281	16%	431	52%	352	**52%**	170	15%	419	12%	386
DP-2x	61%	146	12%	245	51%	247	41%	125	11%	177	7%	243
DP+*DemoSpeedup*	**74%**	**107**	**29%**	**218**	**54%**	**217**	49%	**113**	**38%**	**171**	**21%**	**205**

Method	**Saucepan to Hob**		**Drawers Close**		**Open Trays**		**Flip Cutlery**		**Cupboard Open**		**Averaged**	
	success rate(↑)	episode len(↓)	success rate(↑)	episode len(↓)	success rate(↑)	episode len(↓)	success rate(↑)	episode len(↓)	success rate(↑)	episode len(↓)	success rate(↑)	speedup (↑)
ACT	86%	383	**100%**	119	**100%**	244	**63%**	193	**100%**	146	77%	1.0×
ACT-2x	81%	224	87%	84	93%	149	49%	**121**	96%	103	69%	1.7×
ACT+*DemoSpeedup*	**92%**	**163**	**100%**	**63**	**100%**	**105**	62%	141	**100%**	**81**	**82%**	**2.1×**
DP	**79%**	324	**96%**	114	94%	245	**22%**	175	**100%**	181	55%	1.0×
DP-2x	41%	242	81%	65	86%	157	18%	127	94%	161	45%	1.6×
DP+*DemoSpeedup*	**79%**	**169**	89%	**59**	**96%**	**138**	17%	**98**	**100%**	**103**	**59%**	**1.9×**

Table 1: **Simulation Results.** *DemoSpeedup* achieves remarkable speedup effects while maintaining comparable success rate across different robot platforms and tasks.

4 Experiments

4.1 Simulation Experiments

Compared Methods. We compare the accelerated policies trained with *DemoSpeedup*-accelerated datasets against the same ACT or DP policies trained with the original-speed demonstrations. Additionally, we compare *DemoSpeedup* to more straightforward test-time acceleration [6] that naively downsamples the action chunk during evaluation by 2×, which we call ACT-2× and DP-2×.

Tasks. We consider a total of 11 tasks selected from Aloha [15] and BiGym [34], shown in Fig 3. *For Aloha*, we focus on the tasks with relatively high precision requirements. We select Transfer Cube and Insert, with 50 human demonstrations provided for each task. *For BiGym*, we focus on mobile manipulation and tasks with longer horizons. We improve ACT and DP to have better performance on mobile manipulation tasks by 1) transforming the base action space into position control; 2) replacing the Resnet [35] with Multi-view Vision Transformer [36] in DP to enhance multi-view fusion ability. For fair evaluation, we replay the demonstrations provided in the benchmark and filter out the failed ones [37]. Tasks with extremely low success rate ($< 10\%$) are excluded. A total of 9 BiGym tasks are selected, with different numbers of demonstrations provided in the benchmark and control frequencies ranging from 20Hz to 50Hz.

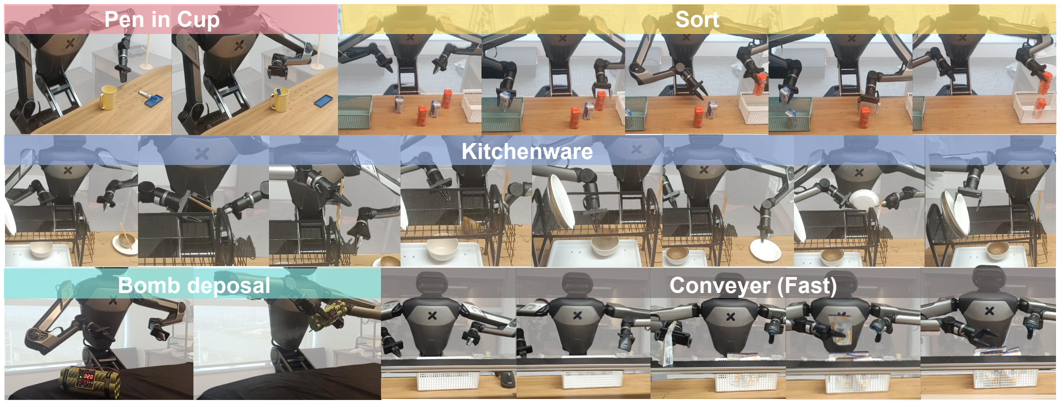

Figure 4: **Real-world Setup.** We consider five real-world challenging tasks. *Sort*, *Kitchenware* emphasize long-horizon manipulation that require multiple skills. *Bomb Deposal* requires precise manipulation. *Conveyer* and its variation *Conveyer Fast* is sensitive to manipulation speed.

Metrics. For evaluation, we report the task completion **success rate** and the averaged **episode length** for a successful policy rollout to measure time efficiency. *For Aloha*, we perform 50 episode rollouts using the checkpoint with minimal validation loss [27, 15]. *For BiGym*, we report the maximum success rate and corresponding average episode length among 50 evaluations throughout the training. All the results are averaged across 3 seeds.

Results. The quantitative results are presented in Table 4. Compared to ACT or DP trained on original-speed demonstrations, the same policies trained with *DemoSpeedup*-accelerated datasets achieve the shortest time to complete the tasks while maintaining comparable performance. Overall, *DemoSpeedup* achieves an average speedup of approximately $2\times$ across different task setups and algorithms, with a maximum speedup of $3\times$. On the other hand, while test-time downsampling shortens the completion time to a certain extent, it causes an average performance drop of over 8%. This reveals the advantage of demonstration acceleration over test-time acceleration.

4.2 Real-World Experiments

Tasks. We design 5 tasks and a variation on Galaxea R1, a bimanual humanoid platform. The tasks emphasize either long horizon or time sensitivity, as illustrated in Figure 4.

- *Pen in Cup*: The robot needs to pick up a pen and place it inside of the cup.
- *Sort*: The robot is required to put all white yoghurt bottles into the green basket and all red ones into the white box.
- *Kitchenware*: The robot needs to grasp the chopsticks, bowl, and plate sequentially with its left arm, transfer them to the right arm, and then place them at the designated location. This is a long-horizon task requiring multiple skills like transferring and insertion.
- *Bomb Deposal*: The robot needs to grasp the bomb toy, move it to its chest, and then precisely collaborate two arms to detach the battery wire.
- *Conveyer*: The robot is required to pick up the scanner, grasp the moving bag on the conveyor belt, scan the bag with the scanner, and then place the bag into the basket. Bags are continuously placed onto the conveyer belt by human.
- *Conveyer Fast*: We evaluate the same checkpoints as in *Conveyer* on a $2\times$ faster conveyer. It aims to simulate the situation where we want the robot more productive than the collected data.

For each task, the RGB visual observations are recorded through a Zed2 Camera mounted on the robot's head, and 100 demonstrations are collected using the GalaxeaVR suite [38]. The object configurations are randomized both in data collection and evaluation.

Method	Pen in Cup		Sort		Bomb Deposal		Kitchenware		Conveyer		Conveyer Fast	
	success rate (↑)	cost time (↓)	success rate (↑)	cost time (↓)	success rate (↑)	cost time (↓)	success rate (↑)	cost time (↓)	success rate (↑)	cost time (↓)	success rate (↑)	cost time (↓)
ACT	$16/30$	$19.45s$	$29/40$	$56.78s$	$\mathbf{7/27}$	$42.13s$	$6/33$	$66.32s$	$18/30$	$13.14s$	$2/30$	$12.68s$
ACT+*Ours*	$\mathbf{24/30}$	**8.28s**	$\mathbf{31/40}$	**20.38s**	$6/27$	**26.31s**	$\mathbf{7/33}$	**27.26s**	$\mathbf{21/30}$	**6.57s**	$\mathbf{16/30}$	**6.28s**
DP	$15/30$	$15.69s$	$32/40$	$39.29s$	$6/27$	$35.69s$	$\mathbf{19/33}$	$61.12s$	$\mathbf{28/30}$	$13.39s$	$7/30$	$12.96s$
DP+*Ours*	$\mathbf{23/30}$	**7.52s**	$\mathbf{38/40}$	**18.32s**	$\mathbf{11/27}$	**19.18s**	$17/33$	**39.23s**	$25/30$	**6.24s**	$\mathbf{27/30}$	**6.03s**

Table 2: **Real-World Results.** The results demonstrate the efficiency of *DemoSpeedup* in accelerating the speed of visuomotor policies and the potential to improve the success rate.

Metrics. We conduct ~ 30 evaluation trials for each task, reporting the number of successful trials and the average time cost. The time cost is recorded by a stopwatch, which starts timing when the robot leaves its default joint positions and stops timing when the robot completes the task and returns to the default joint positions.

Efficiency in boosting the speed of policy. As shown in Table 4.2, *DemoSpeedup* achieves the lowest cost time among different tasks while maintaining the performance. For tasks that require much accuracy such as Bomb Deposal and Kitchenware, *DemoSpeedup* achieves at least 160% speedup. For tasks that are not demanding on precision, *DemoSpeedup* achieves a even higher speedup, such as 278% for ACT and 214% for DP in the Sort task. Besides, we notice that *DemoSpeedup* obtains a higher speedup on ACT than DP. This is partly due to the DP inference delay. The sudden pause caused by the delay between faster movements can make the motions of DP a little more jittery, leading to a slight decrease in acceleration outcome.

Potential for improving the success rate. Interestingly, *DemoSpeedup* could even boost the success rate in some tasks. We argue this is partially because *DemoSpeedup* reduces the decision horizon, thereby reducing the compounding error in imitation learning [29]. Another reason is that when training policy with demonstrations, the change per timestep decreases proportionally as the speed is lower. Thus the marginal information of the action at each timestep is reduced, making it challenging for the policy training to converge and fit complex datasets [39]. For example, in Pen in Cup tasks, the test positions of the cup are covered by the training data, but policies trained on original demonstrations are more likely to miss the correct position of the cup than those trained on speedup demonstrations. In addition, we observe that due to the real-world and Aloha demonstrations being slower than those in Bigym, the performance gains from *DemoSpeedup* are more pronounced in the former two. This indicates that there is some correlation between the quality and speed of data, and *DemoSpeedup* helps for fitting the dataset.

4.3 Ablations

We select two tasks from Aloha and utilize ACT and DP to conduct ablation studies in more details. Three designs are ablated: naively downsample the whole trajectory instead of the replicate-before-downsample(RBD) strategy; adopt the same action chunk length instead of geometrically consistent chunk length; gripper without high-precision control. We report the success rate averaged across tasks. As shown in Table 3, all these designs are significant for the performance of *DemoSpeedup*.

Ablation	ACT	DP
DemoSpeedup	**56%**	**52%**
w/o. RBD strategy	29%	26%
w/o. geometrical consistency	31%	34%
w/o. high precision ctrl	53%	41%

Table 3: **Success rates on ablations.**

4.4 Visualization Analysis

To delve deeper into what patterns the entropy captures, we visualize the entropy curve alongside snapshots from the corresponding demonstrations. As shown in Figure 5, the entropy and our seg-

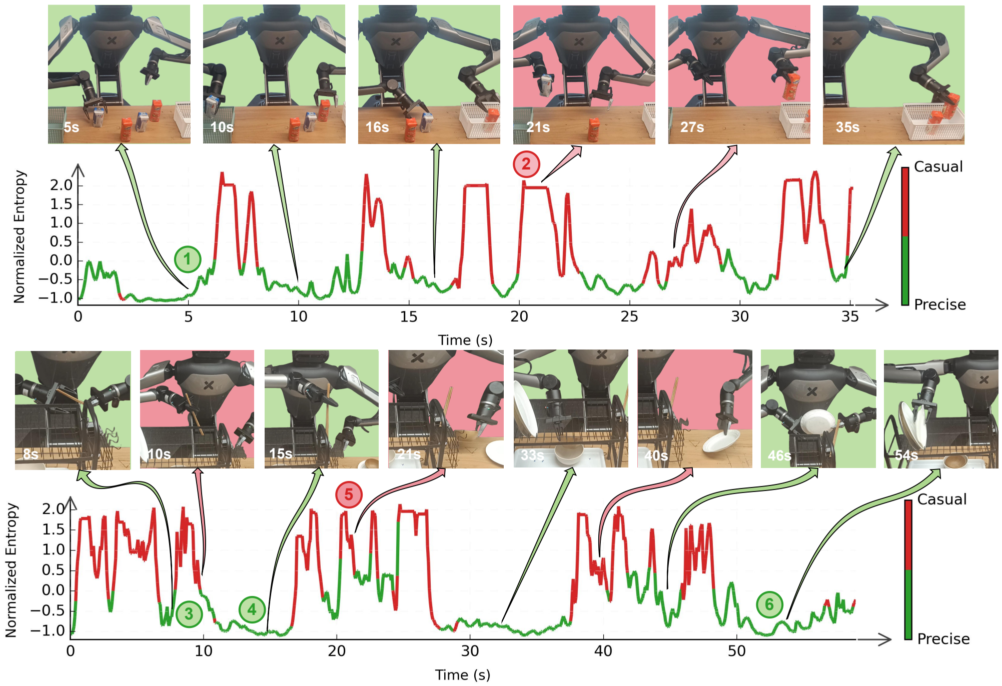

Figure 5: **Entropy Visualization.** We showcase snapshots from the replayed demonstration and the corresponding normalized entropy curve of *Sort*(upper row) and *Kitchenware*(lower row). The green of the curve and the background stands for segmented precision set while red represents casualness set. The estimated entropy effectively captures both delicate skills and causal movements.

ment approach effectively distinguish precise skills from nonchalant movements. Most Motions that approaches an object (Mark 2) or move an object in the air (Mark 5) are recognized as impeccable part. For precision part, the entropy curve could recognize not only contact-rich skills, like picking the yogurt (Mark 1) and transferring the chopsticks (Mark 3), but also contact-free motions such as carefully withdrawing the gripper from the inserted plate to prevent knocking over the plate (Mark 6), or cautiously aligning the chopsticks with the narrow box gap (Mark 4).

5 Conclusion

In this paper, we present *DemoSpeedup*, a self-supervised method to accelerate visuomotor policy execution. *DemoSpeedup* leverages the action entropy of the data estimated from a trained generative policy to guide the acceleration of demonstrations. A clustering-based scheme is proposed to segment the demonstrations into different precision levels according to the entropy. Then those segments are down-sampled at rates that increase with the entropy. Our experiments demonstrate the *DemoSpeedup* can achieve remarkable speedup while maintaining the task performance across different imitation learning algorithms and robot platforms.

Limitations. There are several limitations of this work. First, though *DemoSpeedup* could improve the success rate in some tasks, it occasionally causes minor performance drops, probably because of the dynamics mismatch between the original and accelerated demonstrations. Second, as a self-supervised approach, the *DemoSpeedup* pipeline avoids the trouble of human supervision. However, due to the inherent variations in the execution speed of datasets collected by different human operators, the potential for acceleration also varies. As a result, the desired acceleration rate in *DemoSpeedup* needs to be manually determined. Finally, this work doesn't consider the DP inference delay that has an influence on execution acceleration. This can be solved using distillation methods[40, 41] or flow-based policies[2].

Acknowledgments

We would like to give special thanks to Galaxea Inc. for hardware support and maintenance, Zhenyu Jiao, Ke Dong and Yixiu Li for their technical support on the controller, Ke Sheng and Zhenghao Qi for their advice on VR setup and imitation learning tuning on Galaxea R1. We also thank Kaizhe Hu for initial discussion on the project's direction and Zhecheng Yuan for helping data collection. Tsinghua University Dushi Program supports this project.

References

[1] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[2] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. $\pi 0$: A vision-language-action flow model for general robot control, 2024. *URL https://arxiv. org/abs/2410.24164*.

[3] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, K. Ghasemipour, C. Finn, and A. Wahid. Aloha unleashed: A simple recipe for robot dexterity. *arXiv preprint arXiv:2410.13126*, 2024.

[4] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv preprint arXiv:2403.03954*, 2024.

[5] Z. Xue, S. Deng, Z. Chen, Y. Wang, Z. Yuan, and H. Xu. Demogen: Synthetic demonstration generation for data-efficient visuomotor policy learning. *arXiv preprint arXiv:2502.16932*, 2025.

[6] https://www.figure.ai/news/helix-logistics. Accessed: 2025-2-26.

[7] J. Xie, Z. Wang, J. Tan, H. Lin, and X. Ma. Subconscious robotic imitation learning. *arXiv preprint arXiv:2412.20368*, 2024.

[8] X. Cheng, J. Li, S. Yang, G. Yang, and X. Wang. Open-television: Teleoperation with immersive active visual feedback. *arXiv preprint arXiv:2407.01512*, 2024.

[9] R. Ding, Y. Qin, J. Zhu, C. Jia, S. Yang, R. Yang, X. Qi, and X. Wang. Bunny-visionpro: Real-time bimanual dexterous teleoperation for imitation learning. *arXiv preprint arXiv:2407.03162*, 2024.

[10] Y. Ze, Z. Chen, W. Wang, T. Chen, X. He, Y. Yuan, X. B. Peng, and J. Wu. Generalizable humanoid manipulation with improved 3d diffusion policies. *arXiv preprint arXiv:2410.10803*, 2024.

[11] Z. Fu, T. Z. Zhao, and C. Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.

[12] P. Wu, Y. Shentu, Z. Yi, X. Lin, and P. Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12156–12163. IEEE, 2024.

[13] S. Yang, M. Liu, Y. Qin, R. Ding, J. Li, X. Cheng, R. Yang, S. Yi, and X. Wang. Ace: A cross-platform visual-exoskeletons system for low-cost dexterous teleoperation. *arXiv preprint arXiv:2408.11805*, 2024.

[14] Y. Jiang, R. Zhang, J. Wong, C. Wang, Y. Ze, H. Yin, C. Gokmen, S. Song, J. Wu, and L. Fei-Fei. Behavior robot suite: Streamlining real-world whole-body manipulation for everyday household activities. *arXiv preprint arXiv:2503.05652*, 2025.

[15] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[16] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, et al. \pi_{0.5}: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[17] D. Qu, H. Song, Q. Chen, Y. Yao, X. Ye, Y. Ding, Z. Wang, J. Gu, B. Zhao, D. Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

[18] J. Liu, H. Chen, P. An, Z. Liu, R. Zhang, C. Gu, X. Li, Z. Guo, S. Chen, M. Liu, et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*, 2025.

[19] Q. Bu, J. Cai, L. Chen, X. Cui, Y. Ding, S. Feng, S. Gao, X. He, X. Huang, S. Jiang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

[20] K. Wu, C. Hou, J. Liu, Z. Che, X. Ju, Z. Yang, M. Li, Y. Zhao, Z. Xu, G. Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. *arXiv preprint arXiv:2412.13877*, 2024.

[21] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[22] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[23] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[24] 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.

[25] Q. Li, Y. Liang, Z. Wang, L. Luo, X. Chen, M. Liao, F. Wei, Y. Deng, S. Xu, Y. Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.

[26] A. S. Chen, A. M. Lessing, Y. Liu, and C. Finn. Curating demonstrations using online experience. *arXiv preprint arXiv:2503.03707*, 2025.

[27] L. X. Shi, A. Sharma, T. Z. Zhao, and C. Finn. Waypoint-based imitation learning for robotic manipulation. *arXiv preprint arXiv:2307.14326*, 2023.

[28] S. Belkhale, Y. Cui, and D. Sadigh. Hydra: Hybrid robot actions for imitation learning. In *Conference on Robot Learning*, pages 2113–2133. PMLR, 2023.

[29] S. Belkhale, Y. Cui, and D. Sadigh. Data quality in imitation learning. *Advances in neural information processing systems*, 36:80375–80395, 2023.

[30] L. Lai, A. Z. Huang, and S. J. Gershman. Action chunking as policy compression. 2022.

[31] J. Heer. Fast & accurate gaussian kernel density estimation. In *2021 IEEE Visualization Conference (VIS)*, pages 11–15. IEEE, 2021.

[32] D. Xu, Y. Wang, Y. Meng, and Z. Zhang. An improved data anomaly detection method based on isolation forest. In *2017 10th international symposium on computational intelligence and design (ISCID)*, volume 2, pages 287–291. IEEE, 2017.

[33] L. McInnes, J. Healy, S. Astels, et al. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017.

[34] N. Chernyadev, N. Backshall, X. Ma, Y. Lu, Y. Seo, and S. James. Bigym: A demo-driven mobile bi-manual manipulation benchmark. *arXiv preprint arXiv:2407.07788*, 2024.

[35] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[36] Y. Seo, J. Kim, S. James, K. Lee, J. Shin, and P. Abbeel. Multi-view masked world models for visual robotic manipulation. In *International Conference on Machine Learning*, pages 30613–30632. PMLR, 2023.

[37] Y. S. P. Abbeel. Coarse-to-fine q-network with action sequence for data-efficient robot learning. *Target*, 2(4e9):6e9.

[38] `https://docs.galaxea-ai.com/zh/Guide/R1/R1_VR_Teleop_Usage_Tutorial/`.

[39] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

[40] A. Prasad, K. Lin, J. Wu, L. Zhou, and J. Bohg. Consistency policy: Accelerated visuomotor policies via consistency distillation. *arXiv preprint arXiv:2405.07503*, 2024.

[41] G. Lu, Z. Gao, T. Chen, W. Dai, Z. Wang, W. Ding, and Y. Tang. Manicm: Real-time 3d diffusion policy via consistency model for robotic manipulation. *arXiv preprint arXiv:2406.01586*, 2024.