Waypoint-Based Imitation Learning for Robotic Manipulation

Anonymous Author(s)
Affiliation
Address
email

Abstract: While imitation learning methods have seen a resurgent interest for robotic manipulation, the well-known problem of *compounding errors* continues to afflict behavioral cloning (BC). Waypoints can help address this problem by reducing the horizon of the learning problem for BC, and thus, the errors compounded over time. However, waypoint labeling is underspecified, and requires additional human supervision. Can we generate waypoints automatically without any additional human supervision? Our key insight is that if a trajectory segment can be approximated by linear motion, the endpoints can be used as waypoints. We propose *Automatic Waypoint Extraction* (AWE) for imitation learning, a preprocessing module to decompose a demonstration into a minimal set of waypoints which when interpolated linearly can approximate the trajectory up to a specified error threshold. AWE can be combined with any BC algorithm, and we find that AWE can increase the success rate of state-of-the-art algorithms by up to 25% in simulation and by 4-28% on real-world bimanual manipulation tasks, reducing the decision making horizon by up to a factor of 10. Anonymized supplemental website: https://anoncorl175.github.io/awe-imitation-learning/

Keywords: imitation learning, waypoints, long-horizon

1 Introduction

The simple supervised learning approach of behavioral cloning (BC) has enabled a compelling set of robotic results, from self-driving vehicles [1] to manipulation [2, 3, 4, 5, 6]. However, due to the lack of corrective feedback in the supervised learning approach, errors grow quadratically in the length of the episode for behavior cloning (BC) algorithms [7, 8], colloquially known as the *compounding errors* problem. Waypoints [9, 10, 4, 11] are a relevant proposition in this context: breaking the demonstration into a subset of "key" states reduces the effective length of the decision-making problem, addressing the compounding errors problem while still allowing the use of simple methods such as BC. However, labeling waypoints is both an underspecified problem and requires additional human supervision.

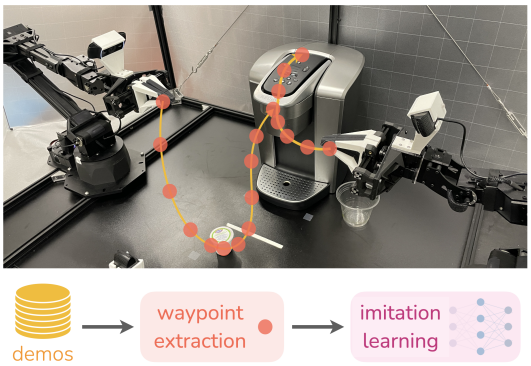

Figure 1: Our approach reduces the horizon of imitation learning by extracting waypoints from demonstrations.

Our objective is to develop a method for selecting waypoints for a given demonstration without any additional human supervision. For a robot arm, if a trajectory segment can be approximated linearly, a low-level controller can reliably imitate the segment without explicitly learning the intermediate states. Thus, we can represent that segment just by the endpoints. Extending this argument to

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

arbitrary trajectories, we can use a subsequence of states as waypoints to represent the trajectory if the trajectory can be approximated well by linearly interpolating between the selected waypoints. The BC prediction problem then transforms from predicting the next state to the next waypoint.

How do we select the waypoint sequence that approximates a given a trajectory? Given a budget for the reconstruction error, where the error is defined as the *maximum* proprioceptive distance between the actual and reconstructed trajectory, we want to select the *shortest* subsequence of states for which the reconstruction error is within budget. This can be posed as a standard dynamic programming problem, by iteratively choosing an intermediate state as the waypoint and recursively selecting waypoints for the two resulting trajectory segments. The recursion terminates whenever the reconstruction error for the linearly interpolated trajectory between the endpoints is already within the budget. Importantly, finding the sequence of waypoints relies only on the robot's *proprioceptive* information, which is already collected during teleoperation. AWE makes no additional assumption about the extrinsic environment (state estimation, point clouds, etc.) and requires no additional label information from humans.

Overall, our work proposes *Automatic Waypoint Extraction* (AWE), a preprocessing module that breaks an expert demonstration into sequence of waypoints. AWE requires minimal additional information, and thus, can easily be plugged into current BC pipelines. We combine AWE with two state-of-the-art imitation learning methods, diffusion policy [5] and action-chunking with transformers (ACT) [6], and study its performance when learning from human-teleoperated demonstrations. On two existing simulated imitation learning benchmarks and multiple real bi-manual manipulation tasks, we find that AWE consistently improves performance, with up to 25% increase in success rate in simulation and 4-28% increase in success rate on real-world tasks.

2 Related Work

Imitation learning is a long-studied approach to training robotic control policies from demonstrations [1, 12, 13]. A central challenge facing imitation learning methods is when compounding errors cause the policy to drift away from states seen in the demonstration data [8], leading to poor performance with small demonstration datasets. Prior approaches have aimed to improve imitation learning performance by developing new policy architectures [14, 11, 15, 16], using policies with expressive distribution classes [17, 18, 5], introducing modified action spaces [19, 20, 21, 22, 4], constructing modified training objectives [2, 6], utilizing particular visual representations [23, 24], incorporating data augmentation [25, 26], or collecting online data [8, 27, 28]. We instead aim to tackle the challenge of compounding errors by extracting waypoints that shorten the horizon. Our approach is orthogonal to and complementary to many of these prior developments; indeed, our experiments show that AWE can be combined with two recent, representative methods [5, 6] to improve their performance.

Similar to our goal, prior works have also aimed to reduce the horizon of imitation learning. Some works reduce the horizon using hand-defined high-level primitive actions [19, 22], whereas we operate on demonstrations collected with high-frequency low-level actions, enabling a range of motions to be demonstrated and avoiding the need to define high-level actions. Other works extract waypoints using heuristics [9, 10, 4, 11], e.g. selecting waypoints as timesteps when the robot is at zero-velocity and actuating the gripper [4]. We are inspired by the success of these methods: they provide dramatic performance and data-efficiency improvements in some settings. However, we find that the heuristics do not apply in general, leading to low success on imitation learning benchmarks and fine manipulation tasks (Sec 5.4). Instead of using heuristics, we extract a set of waypoints that minimize the trajectory reconstruction cost, which yields improvements on two existing simulated manipulation benchmarks and multiple real robot tasks.

3 Preliminaries

Problem Setup. We assume an expert collected dataset of demonstrations $\mathcal{D} = \{\tau_0, \tau_1 \dots \tau_n\}$, where each trajectory $\tau_i = \{(o_j, x_j)\}$ is a sequence of paired raw visual observations o and proprioceptive information x. The proprioceptive information can either be the end-effector pose or joint angles,

and includes the gripper width. In this work, we use position-control for the action space, i.e., proprioceptives x are the action outputs as well. Next, we briefly review two recent successful methods for BC, which we will use in our experiments.

Diffusion Policy. Diffusion policy [5] models the conditional action distribution as a denoising diffusion probabilistic model (DDPM) [29], allowing for better representation of the multi-modality in human-collected demonstrations. Specifically, diffusion policy uses DDPM to model the action sequence $p(A_t \mid o_t, x_t)$, where $A_t = \{a_t, \dots a_{t+C}\}$ represents a chunk of next C actions. The final action is output of the following denoising process [30]:

$$A_t^{k-1} = \alpha \left(A_t^k - \gamma \epsilon_\theta(o_t, x_t, A_t^k, k) + \mathcal{N}(0, \sigma^2 I) \right), \tag{1}$$

where A_t^k is the denoised action sequence at time k. Denoising starts from $\mathcal{A}_t^K$ sampled from Gaussian noise and is repeated till $k = 1$. In Eq 1, (α, γ, σ) are the parameters of the denoising process and ϵ_θ is the score function trained using the MSE loss $\ell(\theta) = (\epsilon_k - \epsilon_\theta(o_t, x_t, A_t^k + \epsilon_k, k))^2$. The noise at step k of the diffusion process, ϵ_k, is sampled from a Gaussian of appropriate variance [29].

Action Chunking with Transformers. Action chunking with transformers (ACT) [6] models the policy distribution $p(A_t \mid o_t, x_t)$ as conditional VAE [31, 32], using a transformer based encoder and decoder. The decoder output is a chunk of actions of size C. Chunking is particularly important for high-frequency fine-grained manipulation tasks, with chunk sizes C being as high as 100 [6].

4 Automatic Waypoint Extraction for Imitation Learning

The goal of this section is to develop our method for Automatic Waypoint Extraction (AWE). First, we define an objective that assesses the quality of the reconstructed trajectory for a given sequence of waypoints. Next, we show how a simple dynamic programming algorithm can be used to select the minimal number of waypoints that have a reconstruction error below a specified threshold. Finally, we discuss how to preprocess a demonstration dataset using AWE before plugging into the BC algorithm, along with the some practical considerations when training and evaluating a waypoint-based policy.

Reconstruction Loss. For an expert demonstration τ, define the sequence of proprioceptives as $\tau_p = \{x_j\}_{j=0}^{|\tau|-1}$ and let $\mathcal{W}$ denote a sequence of waypoints such that $\mathcal{W} = \{w_0, \dots w_L\}$, where w_i denotes the proprioceptive information in the waypoint. We reconstruct an approximate trajectory $\hat{\tau}$ by interpolating between the waypoints, i.e. $\hat{\tau} = f(\mathcal{W})$ for an interpolation function f. While we restrict to linear interpolation in this work, the framework can be extended to incorporate splines.

To measure how well a sequence of waypoints approximates the true trajectory, we want to measure how much the interpolated trajectory deviates from the true trajectory. We define the reconstruction loss as the maximum distance of any state in the original trajectory from the reconstructed trajectory, that is,

$$\mathcal{L}(\hat{\tau}, \tau) = \max_{x \in \tau_p} \min_{\hat{x} \in \hat{\tau}} \ell(x, \hat{x}) \tag{2}$$

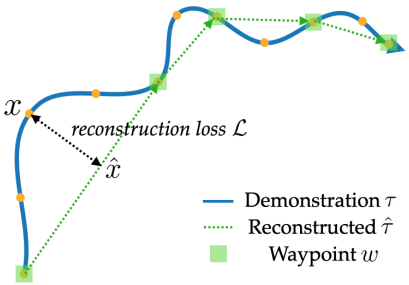

Figure 2: Visualizing the loss $\mathcal{L}$.

where $\ell(\cdot, \cdot)$ is some distance function (for example, Euclidean ℓ_2 distance). The $\min_{\hat{x} \in \hat{\tau}} \ell(\cdot, \hat{x})$ denotes shortest distance of a proprioceptive state to the interpolated trajectory $\hat{\tau}$. How do we aggregate projection errors for proprioceptives in the true trajectory? While there are several options, for example, mean projection error over τ_p, we define the reconstruction loss as the maximum projection error over all proprioceptives in τ_p. The success of a trajectory often relies on reaching key states, and the mean error can be low while having a high projection error for those key states. While a low reconstruction loss with maximum projection error also does not guarantee downstream success, it encourages minimizing outlier projection errors potentially critical for a successful execution. The reconstruction loss $\mathcal{L}$ is visualized in Figure 2.

Waypoint Selection via Dynamic Programming. Given the reconstruction loss, how do we use it to optimize waypoints? We consider the following optimization problem:

$$\min_{\mathcal{W}} |\mathcal{W}| \quad \text{s.t.} \quad \mathcal{L}(f(\mathcal{W}), \tau) \leq \eta, \tag{3}$$

i.e., minimize the number of selected waypoints such that the reconstruction loss is below the budget η. As presented, waypoints can be arbitrary points in the proprioceptive space, but we will restrict waypoint selection to the states visited in the expert trajectory τ. The problem simplifies to finding the shortest subsequence of τ_p such that the reconstruction loss is less than η, which can be solved efficiently with dynamic programming (DP). For a trajectory segment, either linearly interpolating between the endpoints sufficiently reconstructs the segment (i.e, reconstruction loss less than η), in which case the endpoints are returned as waypoints. Or for every intermediate state between the endpoints: (1) break the trajectory into two segments at that intermediate state and, (2) recursively find the shortest subsequence for each segment. Finally, choose the intermediate state resulting in the shortest subsequence when the waypoints from its two trajectory segments are merged, and return the merged waypoints. The pseudocode for selecting waypoints with DP is in Algorithm 1.

Preprocessing Demonstrations. For an expert trajectory $\tau = \{(o_0, x_0), \dots (o_T, x_T)\}$, denote the selected waypoints as $\mathcal{W} = \{(w_0, t_0) \dots (w_L, t_L)\}$, where w_i denotes the waypoint and t_i denotes the time index in τ. The training problem for a BC algorithm changes from predicting the next proprioceptive state to next waypoint. However, if done naïvely, the training dataset of next waypoints will be much smaller. But, we can use all observations in τ between two consecutive waypoints by labeling them with closest waypoint after the observation. This follows from the intuition that following the waypoints implies the robot tries to reach w_{k+1} from w_k, and therefore, should target w_{k+1} from intermediate states between them as well. The final dataset can be written as $\mathcal{D}^{\tau}_{\text{waypoint}} = \{(o_t, x_t, w_{\texttt{next_wp}(t)})\}_{t=0}^{T-1}$ where $\texttt{next_wp}(t) = \text{argmin}_{j \in \{0,1,\dots L\}}$ such that $t_j > t$. The process of selecting waypoints and constructing the augmented dataset is repeated for every expert demonstration $\tau \in \mathcal{D}$, and the resulting datasets are merged to get the final training dataset.

Overview and Practical Considerations. We have proposed AWE, a simple method that can preprocess a demonstration into sequence of waypoints without any additional supervision. The training dataset can be relabeled with the next waypoint instead of next propriopceptive state, and plugged into a BC pipeline. The choice of policy distribution class used with AWE is important; waypoints introduce additional multi-modality into the conditional action distribution as different demonstrations may be processed into different waypoints. Using more expressive policy classes capable of representing multi-modal action distributions is critical, as introducing waypoints can make the performance *worse* for less expressive policy classes (Figure 6).

Why does AWE return meaningful waypoints? An intuitive notion of waypoints relies on registering important events happening in the extrinsic environment (grasping a cup, opening a door, etc.) while AWE uses just the proprioceptive information to select waypoints. AWE relies on the idea that the expert demonstrations will naturally deviate from linear motion during such key events. For simpler parts of the task, such as free-space reaching, demonstrations are more likely to be approximated by linear motion, resulting in fewer waypoints. Moreover, decreasing the budget η allows for selecting more waypoints in general, and thus, better reconstruction as visualized in Figure 4.

An important consideration at test-time is to allow more time for position-control to reach waypoints, as waypoints are farther apart compared to proprioceptive positions in the original expert demonstrations. The exact instantiation for the low-level controller depends on the whether the robot is operating in the joint space or end-effector space, which we discuss in Appendix C.1.

5 Experiments

Our experiments seek to answer the following questions: (1) How well does AWE combine with representative behavioral cloning methods? (2) Can it be used to tackle standard imitation learning benchmarks with real human demonstrations? (3) Can it be effective on a real-robot? (4) How does the parameterization of the policy affect the performance? (5) How do the selected waypoints and downstream performance change as we vary the hyperparameters? To answer these questions,

Table 1: **Success rate (%) for simulated bimanual tasks.** We report results on both training with scripted data and training with human data, with 3 seeds and 50 policy evaluations each. Baseline results are obtained from Zhao et al. [6]. Overall, AWE +ACT significantly outperforms previous methods.

	Cube Transfer		**Bimanual Insertion**	
	scripted data	human data	scripted data	human data
BC-ConvMLP	1	0	1	0
BeT [14]	27	1	3	0
RT-1 [15]	2	0	1	0
VINN [24]	3	0	1	0
ACT [6]	86	50	32	20
AWE +ACT (Ours)	**99**	**71**	**57**	**30**

we compare the performance of recent state-of-the-art BC methods with and without AWE on 8 tasks and 10 datasets. First, we evaluate AWE on a set of simulation environments, specifically two bimanual manipulation tasks from Zhao et al. [6] and three manipulation tasks from the RoboMimic benchmark [17]. We evaluate AWE on a set of three bimanual manipulation tasks on the real robot: *coffee making*, *wiping the table* and *screwdriver handover*. Hyperparameter and implementation details can be found in Appendix B and C respectively.

5.1 Bimanual Simulation Suite

The bimanual simulation suite contains two fine-grained long-horizon manipulation tasks in MuJoCo [33]. The observation space includes a 480×640 image and the current joint positions for both robots. The 14-dimensional action space corresponds to the target joint positions. Demonstrations are 400 to 500 steps at a control frequency of 50Hz. In the **Cube Transfer** task, the right robot arm needs to pick up the cube from a random position on the table, and then hand it to the left arm mid-air. For **Bimanual Insertion**, both the peg and the socket are placed randomly on the table. The arms need to first pick them up respectively, then insert the peg into the socket mid-air. Both tasks require delicate coordination between the two arms and closed-loop visual feedback: error in grasping can directly lead to failure of handover or insertion.

Two datasets are available for each task: one collected with a scripted policy, and one collected by human demonstrators, both with 50 demonstrations. As shown in Table 1, AWE outperforms competitive BC baselines on all tasks and datasets in the bimanual simulation suite, where some of the baselines completely fail due to task difficulty. AWE can increase the success rate of ACT, the state-of-the-art method on this benchmark 17% on an average, and up to 25% on the scripted bimanual insertion. The effective length of the training demonstrations reduce by a factor of $7\times$ to $10\times$, even allowing for improvements on human data which is fairly multi-modal to begin with. Notably, the performance improves by 50% when imitating human demonstrations on the bimanual insertion task. Overall, this suggests that the benefit from reducing the effective training horizon exceeds any potential downside from the increased multi-modality introduced by AWE.

5.2 RoboMimic Suite

Next, we evaluate on three simulated tasks from the **RoboMimic** [17] manipulation suite: **Lift** where the robot arm has to pick up a cube from the table, **Can** where the robots are required to pick up a soda can from a large bin and place it into a smaller target bin, and **Square** where robots are tasked to pick up a square nut and place it on a rod. It is challenging due to the high precision needed to pick up the handle and insert it into a tightly-fitted rod. Episodes start with randomly initialized object configuration and terminate upon successfully completing the task or exceeding pre-defined horizons. All environments return RGB observations and the action space is the 6DoF end-effector pose, with an additional degree for the gripper.

We combine AWE with the state-of-the-art method on this benchmark, Diffusion Policy [5]. Since, diffusion policy achieves a near-perfect success rates on **Lift**, **Can**, and **Square** when training on a dataset with 200 proficient-human demonstrations, we focus our evaluation on how the performance

Table 2: **Success rate (%) for behavior cloning benchmark, RoboMimic (Visual Policy).** We evaluate the policy every 100 epochs across the training, and report the average of the max performance across 3 training seeds and 30 different environment initial conditions (90 in total). Results on LSTM-GMM and IBC are obtained from Chi et al. [5] for comparison to more traditional methods.

Task	# Demos	AWE + Diffusion (Ours)	Diffusion	LSTM-GMM	IBC
Lift	30	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	-	-
	50	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	-	-
	100	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	-	-
	200	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	96	73
Can	30	$\mathbf{69.0 \pm 1.4}$	61.0 ± 5.9	-	-
	50	$\mathbf{85.7 \pm 1.9}$	82.3 ± 3.3	-	-
	100	$\mathbf{95.3 \pm 1.7}$	93.3 ± 0.0	-	-
	200	$\mathbf{96.7 \pm 0.9}$	$\mathbf{97.3 \pm 2.5}$	88	1
Square	30	$\mathbf{62.3 \pm 3.3}$	44.3 ± 6.1	-	-
	50	$\mathbf{67.0 \pm 2.9}$	57.3 ± 4.2	-	-
	100	$\mathbf{91.7 \pm 3.9}$	82.0 ± 7.0	-	-
	200	$\mathbf{94.7 \pm 3.9}$	$\mathbf{95.0 \pm 4.1}$	59	0

scales with the number of demonstrations, both with and without AWE. The results in Table 2 suggest that AWE consistently improves the performance of diffusion policy as the number of demonstrations is scaled from 30 to 200, while both of them outperform LSTM-GMM and implicit BC [34] with half the demonstration data, or even less. The improvements are larger when the number of demonstrations is smaller or the task is longer-horizon, for example, an 18% increase in the success rate when using 30 demonstrations on the **Square** task.

5.3 Real-World Bimanual Tasks

For real-robot evaluations, we use ALOHA [6], a low-cost open-source bimanual hardware setup. The setup consists of two leader arms and two follower arms, where the joint positions are synchronized between the leaders and followers during teleoperation. The observation space consists of RGB images from 4 cameras: two are mounted on the wrist of the follower robots, allowing for close-up views of objects for fine-manipulation, and the other two are mounted on the front and at the top respectively. The demonstration data consists of 4 camera streams and the joint positions for each robot at 50Hz. We refer readers to the original paper for more hardware details.

We experiment with three long-horizon tasks, each requiring precise coordination between the two arms, illustrated in Figure 3. For **Screwdriver Handover**, the right arm needs to pick up the screwdriver that is randomly initialized in a 15cm $\times$20cm rectangular region (#1) and hand it to the left arm mid-air (#2), followed by the left arm dropping it into the cup (#3). For **Wiping the Table**, a roll of paper towels is randomly placed in a 15cm $\times$10cm region. The opening of the roll always faces the right side, with naturally occurring variations in length and spacing. The left arm presses on the roll to prevent it from moving (#1), while the right arm tears off one segment of the paper towel (#2) and places it on a fixed location to absorb the spilled liquid (#3). For **Coffee Making**, a small coffee pod is randomized in a 15cm $\times$10cm region. The left arm needs to pick it up (#1), followed by the right arm opening the coffee machine (#2). The left arm then carefully inserts the coffee pod into the slot (#3), with the right arm closing the lid (#4). Next, the right arm grasps a transparent cup with upto 2cm randomization in the position and places it under the coffee outlet (#5).

The three tasks emphasize precision and coordination, and involve deformable or transparent objects that can be hard to perceive or simulate. For example, placing the coffee pod into the machine requires high precision. It is easy for the gripper or coffee pod to collide with the machine due to the small clearance. The screwdriver handover task emphasizes the coordination between two arms. Grasping the paper towel requires accurate perception of the deformable material, which also has low-contrast against itself. The gripper needs to move accurately so as to only grasp the opening but not collide with the roll and push it away.

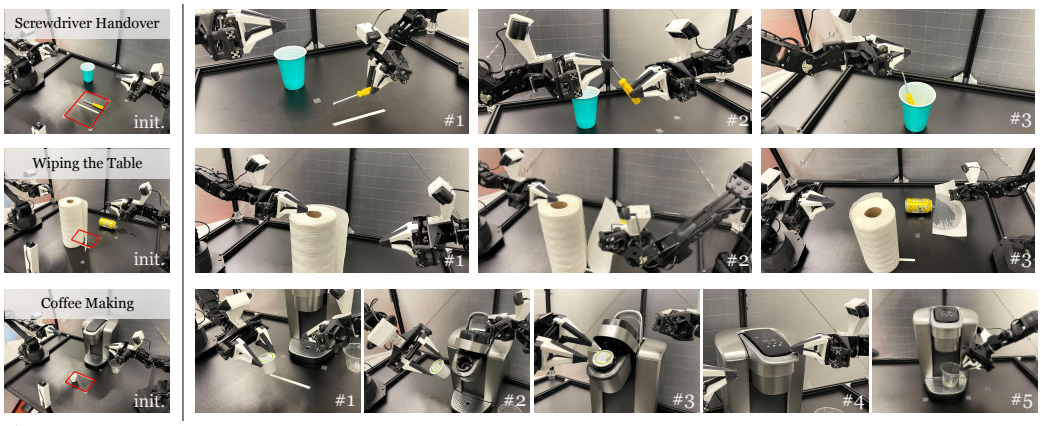

Figure 3: **Real-World Bimanual Tasks.** We consider three challenging real-world bi-manual tasks: (top) picking up a screw driver, handing it over to the other arm, and placing it in a cup, (middle) tearing off a segment of paper towel and putting it on a spill, and (bottom) putting a coffee pod into a coffee machine, closing the coffee machine, and placing a cup underneath the dispenser. The initial object positions are initialized within the red rectangle.

Table 3: **Success rate (%) for real world tasks.** Our automatic waypoint extraction method improves the success of ACT on all three tasks. On the most difficult coffee making task, AWE improves success by 28%.

	Screwdriver Handover	Wiping the Table	Coffee Making
ACT	84	92	36
AWE + ACT (Ours)	**92**	**96**	**64**

As shown in Table 3, AWE achieves substantial success on each task. It consistently improves over ACT by 8%, 4%, and 28% on Screwdriver handover, Wiping table, and Coffee Making, respectively. We observe that the most common failure case for ACT is inaccurate action prediction, which results from compounding errors on these long-horizon tasks. For example, the robot may make a wrong prediction at the beginning and grasp the coffee pod at an inconvenient position. The subsequent predictions become increasingly incorrect, and thus the robot fails to insert the coffee pod into the machine. On the other hand, AWE can more accurately grasp the coffee pod due to a smaller decision horizon, resulting in more successful insertions into the coffee machine. Leveraging the low-level controller to execute linear motions instead of relying on accurate policy predictions can reduce the errors compounded over time. By accurately detecting waypoints for a successful handover, for tearing, and for inserting, AWE decreases the policy horizon and consistently improves performance.

5.4 Analysis

Waypoint selection for different error budgets. We visualize a ground truth trajectory of end-effector positions and the EE trajectory reconstructed using AWE for the **Can** task in Figure 4. As the the error budget η is reduced, the reconstructed trajectory tracks the original trajectory better. Importantly, as the budget is decreased, waypoints are added to harder segments of the task, as they are less linear while the number of waypoints for simpler, linear paths stays similar. Smaller error thresholds lead to gradual increases in the number of selected waypoints. We also measure performance with varying error thresholds (the only hyperparameter), for AWE +DiffusionPolicy on the Can task with 50 demonstrations. Figure 5 shows that when the η is too high (too few waypoints) or too low (too many waypoints), the agent does not take full advantage of AWE.

On the importance of modeling multi-modality for AWE. The usage of waypoints can increase the multimodality of the target conditional action distribution. We compare the performance of AWE when trained with mean-squared error (MSE) loss (i.e, a unimodal Gaussian with identity covariance) and a more expressive Gaussian mixture model (GMM) with 5 modes. As shown in Figure 6, GMM policies can benefit from AWE, as they can represent multimodal action distributions. However, vanilla BC with a MSE loss degrades in performance. BC has a mode-covering behavior, and insufficient representative power of unimodal Gaussian can cause the performance to degrade.

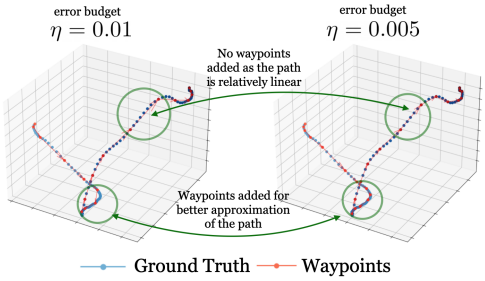

Figure 4: Progression of waypoints selected by AWE as the error budget η reduces. Fewer waypoints are added if the segment is better approximated by linear interpolation.

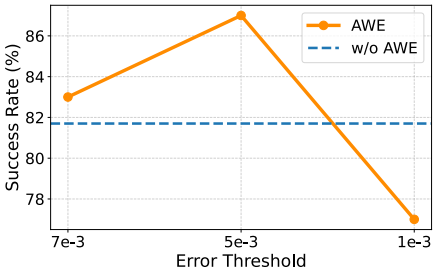

Figure 5: Success rate vs. error budget threshold η. Performance drops slightly if the budget is too tight and more significantly if the budget is too permissive.

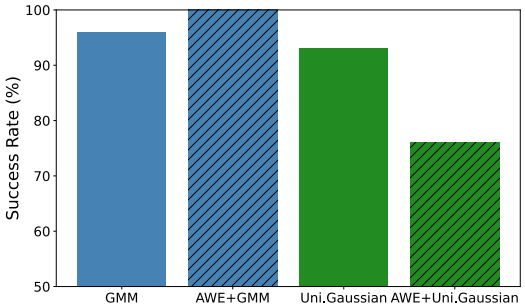

Figure 6: While expressive policies that can represent multimodal distributions benefit from AWE (GMMs on *left*), the performance can degrade for policy classes that are not sufficiently expressive (*right*).

Method	**Replay Success (%)**
Velocity + Gripper [4]	5.5
Constant	32.0
AWE	99.5

Figure 7: Comparing AWE to different heuristics for waypoint selection, when replaying trajectories on the simplest **Lift** task in RoboMimic. Following the waypoints extracted with heuristics (e.g. zero-velocity states and gripper action states, constant-interval states) leads to poor task completion, whereas following waypoints from AWE can consistently complete the task.

Comparison to heuristics. Prior works [9, 10, 4, 11] have been successful by extracting waypoints using simple heuristics. Are simple heuristics enough for extracting important waypoints? We experiment with two heuristics. The first one is similar to Shridhar et al. [4], labeling timesteps as waypoints when the end-effector velocity is close to zero, or when the binary gripper state changes. The second heuristic selects waypoints with fixed intervals (here we used 15). For AWE and both heuristics, we extract waypoints for all 200 trajectories in the RoboMimic **Lift** dataset, and measure the success rate when replaying the demonstration, i.e. following the extracted waypoints starting from the demonstration trajectory's initial state. Results in Figure 7 show that these two heuristics do not lead to satisfactory success rates even when simply replaying the trajectories.

6 Conclusion

We presented a method for extracting waypoints from demonstrations of robotic manipulation tasks, therefore reducing the horizon of imitation learning problems. We found that our automatic waypoint extraction method (AWE) can be combined with state-of-the-art imitation learning methods such as diffusion policy and ACT to improve performance, especially in data limited settings where the task performance is not already saturated. AWE also consistently improved performance on three real-world dexterous manipulation tasks. Finally our analysis indicated the importance of the AWE optimization compared to naive or heuristic waypoint selection methods, as well as the effect of the error budget and policy distribution class on performance.

Limitations. AWE leverages proprioceptive information to reparameterize demonstration trajectories in end-effector or joint space, an approach that may not be applicable to torque-controlled robot arms, tasks requiring forceful manipulation, or other robotics problems such as purely visual navigation or legged locomotion. Furthermore, for tasks that require a great deal of precision at certain times, we expect that AWE would require a tight error budget, which would in turn reduce the effect of using waypoints compared to the original demonstration. This limitation of AWE might be resolved by identifying when such precision is needed, either automatically or by incorporating some human supervision, and subsequently modifying the AWE optimization objective.

References

[1] D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

[2] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5628–5635. IEEE, 2018.

[3] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser. Tossingbot: Learning to throw arbitrary objects with residual physics. *IEEE Transactions on Robotics*, 36(4):1307–1319, 2020.

[4] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[5] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[6] 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.

[7] S. Ross and D. Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668. JMLR Workshop and Conference Proceedings, 2010.

[8] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[9] K. Hsiao and T. Lozano-Perez. Imitation learning of whole-body grasps. In *2006 IEEE/RSJ international conference on intelligent robots and systems*, pages 5657–5662. IEEE, 2006.

[10] B. Akgun, M. Cakmak, K. Jiang, and A. L. Thomaz. Keyframe-based learning from demonstration: Method and evaluation. *International Journal of Social Robotics*, 4:343–355, 2012.

[11] S. James and A. J. Davison. Q-attention: Enabling efficient learning for vision-based robotic manipulation. *IEEE Robotics and Automation Letters*, 7(2):1612–1619, 2022.

[12] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3 (6):233–242, 1999.

[13] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[14] N. M. Shafiullah, Z. Cui, A. A. Altanzaya, and L. Pinto. Behavior transformers: Cloning k modes with one stone. *Advances in neural information processing systems*, 35:22955–22968, 2022.

[15] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[16] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors. *arXiv preprint arXiv:2210.11339*, 2022.

[17] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *arXiv preprint arXiv:2108.03298*, 2021.

[18] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson. Implicit behavioral cloning. In *Conference on Robot Learning*, pages 158–168. PMLR, 2022.

[19] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pages 726–747. PMLR, 2021.

[20] C. Wang, R. Wang, A. Mandlekar, L. Fei-Fei, S. Savarese, and D. Xu. Generalization through hand-eye coordination: An action space for learning spatially-invariant visuomotor control. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8913–8920. IEEE, 2021.

[21] E. Johns. Coarse-to-fine imitation learning: Robot manipulation from a single demonstration. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 4613–4619. IEEE, 2021.

[22] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[23] P. Florence, L. Manuelli, and R. Tedrake. Self-supervised correspondence in visuomotor policy learning. *IEEE Robotics and Automation Letters*, 5(2):492–499, 2019.

[24] J. Pari, N. M. Shafiullah, S. P. Arunachalam, and L. Pinto. The surprising effectiveness of representation learning for visual imitation. *arXiv preprint arXiv:2112.01511*, 2021.

[25] M. Jia, D. Wang, G. Su, D. Klee, X. Zhu, R. Walters, and R. Platt. Seil: Simulation-augmented equivariant imitation learning. *arXiv preprint arXiv:2211.00194*, 2022.

[26] A. Zhou, M. J. Kim, L. Wang, P. Florence, and C. Finn. Nerf in the palm of your hand: Corrective augmentation for robotics via novel-view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17907–17917, 2023.

[27] M. Laskey, S. Staszak, W. Y.-S. Hsieh, J. Mahler, F. T. Pokorny, A. D. Dragan, and K. Goldberg. Shiv: Reducing supervisor burden in dagger using support vectors for efficient learning from demonstrations in high dimensional state spaces. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 462–469. IEEE, 2016.

[28] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[29] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[30] M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688, 2011.

[31] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[32] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

[33] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012. doi:10.1109/IROS.2012.6386109.

[34] P. R. Florence, C. Lynch, A. Zeng, O. Ramirez, A. Wahid, L. Downs, A. S. Wong, J. Lee, I. Mordatch, and J. Tompson. Implicit behavioral cloning. *Conference On Robot Learning*, 2021.