Enabling Stateful Behaviors for Diffusion-based Policy Learning

Xiao Liu, Fabian Weigend, Yifan Zhou, and Heni Ben Amor

Abstract— While imitation learning provides a simple and effective framework for policy learning, acquiring consistent actions during robot execution remains a challenging task. Existing approaches primarily focus on either modifying the action representation at data curation stage or altering the model itself, both of which do not fully address the scalability of consistent action generation. To overcome this limitation, we introduce the Diff-Control policy, which utilizes a diffusion-based model to learn the action representation from a state-space modeling viewpoint. We demonstrate that we can reduce diffusion-based policies' uncertainty by making it stateful through a Bayesian formulation facilitated by ControlNet, leading to improved robustness and success rates. Our experimental results demonstrate the significance of incorporating action statefulness in policy learning, where Diff-Control shows improved performance across various tasks. Specifically, Diff-Control achieves an average success rate of 72% and 84% on stateful and dynamic tasks, respectively. Project page: `https://github.com/ir-lab/Diff-Control`

I. INTRODUCTION

Previous studies have explored various approaches to learning behavioral cloning policies, such as directly outputting actions via regression models [1] or utilizing implicit policies [2]. Notably, diffusion-based policies [3] have emerged as a standout choice due to their ability to model multimodal action distributions effectively, leading to enhanced performance.

In practice however, the concern over inconsistency in action representation remains a persistent challenge. Such inconsistencies can lead to noticeable disparities between the distribution of robot trajectories and the underlying environment, thereby limiting the efficacy of control policies [4]. The primary causes of this inconsistency typically stem from the context-rich nature of human demonstrations [5], distribution shift problems [6], and the volatile nature of high-dynamic environments. Previous approaches, such as action chunking [7] and predicting closed-loop action sequences [3], have been proposed to address this issue. Additionally, Hydra [8] and Waypoint-based manipulation [9] modify action representations to ensure consistency. However, these approaches address the problem by altering the action representation without using the actions as is.

Instead, can we learn to explicitly impose temporal consistency by incorporating temporal transitions within diffusion policies? In the realm of deep state-space models (DSSMs) [10]–[12], the effective learning of a state transition model enables the identification of underlying dynamic patterns. In this paper, we argue that such deep transition

All authors are with the School of Computing and Augmented Intelligence, Arizona State University, USA {xliu330, fweigend,, yzhou298, hbenamor}@asu.edu
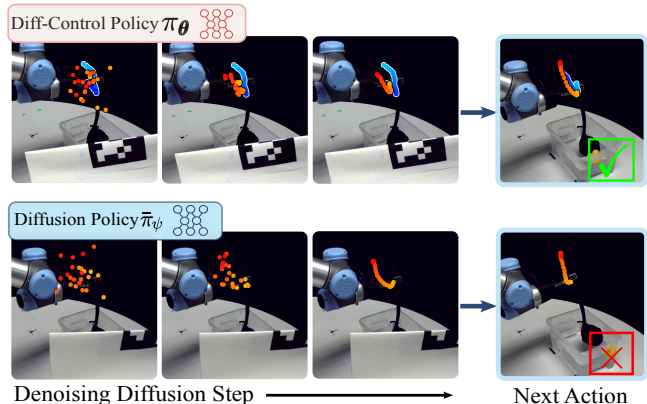
Fig. 1. **Diff-Control Policy** incorporates ControlNet, functioning as a transition model that captures temporal transitions within the action space to ensure action consistency.

models can easily be integrated into diffusion policies in order to capture temporal action dynamics as a state-space model. This integration makes the policy stateful, thereby increasing robustness and success rates.

We propose **Diff-Control**, a stateful diffusion-based policy that generates actions and enables learning an action transition model concurrently. Building upon the ControlNet framework introduced by [13] for spatial conditioning control in image generation, we leverage it as the transition model to provide temporal conditioning to a base diffusion policy. As shown in Figure 1, a prior action sequence (in blue) is utilized as condition when generating new action sequence (in red). The main contributions of the paper are:

- A deep, recursive Bayesian filter within diffusion-based polices using ControlNet structure as a transition model to ensure consistent action generation.
- Diff-Control stateful policy representation performing dynamic and temporal sensitive tasks with at least 8% and 48% improvement in success rate.
- Diff-Control policy exhibits notable precision and robustness against perturbations, achieving at least a 30% higher success rate compared to state-of-the-art methods.

II. METHOD

The key objective of Diff-Control is to learn how to incorporate state information into the decision-making process of diffusion policies. An illustrative example for this behavior is shown in Figure 2: a policy learning to approximate a cosine function. Given single observation at time t, stateless policies encounter difficulties in producing accurate generating the continuation of trajectories. Due to ambiguities, Diffusion
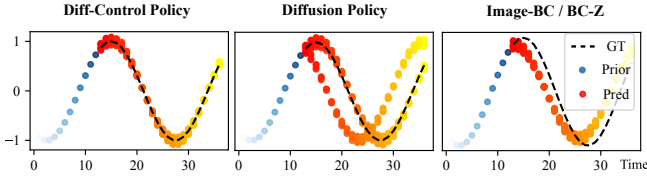
Fig. 2. **Stateful behavior**: at a given state, Diff-Control policy can utilize prior trajectories to approximate the desired function. Diffusion policy [3] learns both modes but fails on generating the correct trajectory cosistently, Image-BC/BC-Z [1] fails to generate the correct trajectory.

Policy [3] tends to learn multiple modes. By contrast, Diff-Control integrates temporal conditioning allowing it to generate trajectories by considering past states. To this end, the proposed approach leverages recent ControlNet architectures to ensure temporal consistency in robot action generation.

In computer vision, ControlNet is used within stable diffusion models to enable additional control inputs or extra conditions when generating images or video sequences. Our method extends the basic principle of ControlNet from image generation to action generation, and use it as a state-space model in which the internal state of the system affects the output of the policy in conjunction with observations (camera input) and human language instructions.

A. Recursive Bayesian Formulation

The objective of our method is to learn a policy with conditions $\mathbf{c}$ and observations $\mathbf{o}$ as input. In this context, we define $\mathbf{a}$ as the trajectory comprising the robot's end-effector pose. In alignment with prior approaches [3], [14], our aim is also to take multiple conditions as input. However, as mention in Section I, efforts have been made to explore robust action generation in previous works [3], [7], [8], they have not accounted for the statefulness of $\mathbf{a}$. We address the action consistency from a Bayesian perspective by introducing transition in action spaces, our formulation is as follows:

$$
\begin{aligned}
p(\mathbf{a}_t|\mathbf{a}_{1:t-1}, \mathbf{o}_{1:t}, \mathbf{c}) \\
\propto p(\mathbf{o}_t|\mathbf{a}_t, \mathbf{c})\, p(\mathbf{a}_t|\mathbf{a}_{1:t-1}, \mathbf{o}_{1:t-1}, \mathbf{c}).
\end{aligned} \tag{1}
$$

Let $\mathrm{bel}(\mathbf{a}_t) = p(\mathbf{a}_t|\mathbf{a}_{1:t-1}, \mathbf{o}, \mathbf{c})$, applying the Markov property, i.e., the assumption that the next generated trajectory is dependent only upon the current trajectory, yields:

$$
\mathrm{bel}(\mathbf{a}_t) = \eta \underbrace{p(\mathbf{o}_t|\mathbf{a}_t, \mathbf{c})}_{\text{observation model}} \prod_{t=1}^{t} \overbrace{p(\mathbf{a}_t|\mathbf{a}_{t-1}, \mathbf{c})}^{\text{transition model}} \mathrm{bel}(\mathbf{a}_{t-1}), \tag{2}
$$

where η is a normalization factor, $p(\mathbf{o}_t|\mathbf{a}_t, \mathbf{c})$ is the observation model and $p(\mathbf{a}_t|\mathbf{a}_{t-1}, \mathbf{c})$ is the transition model.

B. Diff-Control Policy

We now show how Bayesian formulation and diffusion model can be coupled together such that one policy can generate stateful action sequences that facilitate consistent robot behaviors. We propose Diff-Control Policy $\pi_{\boldsymbol{\theta}}(\mathbf{a}_{[W_t]}|\mathbf{o}, \mathbf{a}_{[W_{t-h}]}, \mathbf{c})$, which is parameterized by $\boldsymbol{\theta}$. Here, h stands for the execution horizon, $\mathbf{c}$ represents a language condition in the form of a natural human instruction, and $\mathbf{o}$
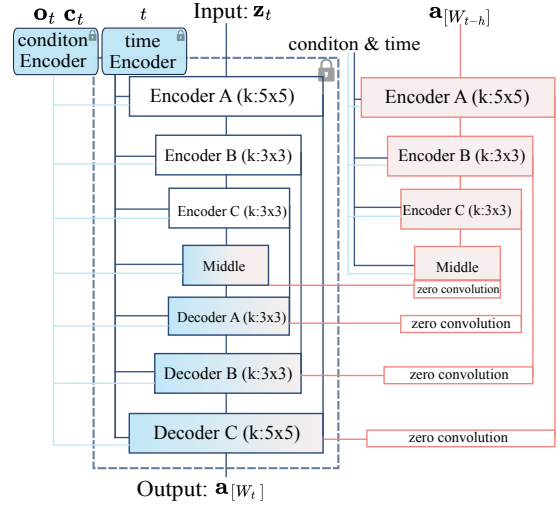

Fig. 3. The Diff-Control Policy is implemented through the utilization of a locked U-net diffusion policy architecture. It replicates the encoder and middle blocks and incorporates zero convolution layers.

denotes a sequence of images captured by an RGB camera of the scene. The policy $\pi_{\boldsymbol{\theta}}$ generates a window of trajectory $\mathbf{a}_{[W_t]} = [\mathbf{a}_1, \mathbf{a}_2, \cdots \mathbf{a}_W]^T \in \mathbb{R}^{7 \times W}$, where W refers to the window size or the prediction horizon.

The Diff-Control policy within the Bayesian formulation comprises two crucial modules. The transition module receives the previous action $\mathbf{a}_{[W_t]}$ and generates latent embeddings for the subsequent utilization by the base policy. Acting as the observation model, the base policy incorporates the temporal information associated with $\mathbf{a}_{[W_t]}$ and produces a new action $\mathbf{a}_{[W_{t+h}]}$. This two-module structure enables the Diff-Control policy to adeptly capture temporal dynamics and facilitate the generation of subsequent actions with accuracy and consistency.

Base Policy: To begin, we train a diffusion-based policy [3] as the base policy $\bar{\pi}_{\boldsymbol{\psi}}(\mathbf{a}_{[W_t]}|\mathbf{o}, \mathbf{c})$. We adopt the 1D temporal convolutional networks from [15] and construct the U-net backbone. The policy $\bar{\pi}_{\boldsymbol{\psi}}$ can autonomously execute and generate actions without any temporal information dependency.

Transition Model: The proposed framework incorporates ControlNet as the Transition Module (depicted in Figure 3). This utilization extends the capability of the policy network to include temporal conditioning effectively. To achieve this, we utilize the previously generated action sequences as the prompt input to ControlNet. By doing so, the base policy $\bar{\pi}_{\boldsymbol{\psi}}$ becomes informed about the previous actions $\mathbf{a}_{[W_{t-h}]}$. We implement ControlNet by creating a trainable replica of the $\bar{\pi}_{\boldsymbol{\psi}}$ encoders and then freeze the base policy $\bar{\pi}_{\boldsymbol{\psi}}$. The trainable replica is connected to the fixed model with zero convolutional layers [16]. ControlNet can then take $\mathbf{a}_{[W_{t-h}]}$ as the conditioning vector and reuses the trained base policy $\bar{\pi}_{\boldsymbol{\psi}}$ to construct the next action sequence $\mathbf{a}_{[W_t]}$.

C. Training

The training process for the base policy $\bar{\pi}_{\boldsymbol{\psi}}(\mathbf{a}_{[W_t]}|\mathbf{o}, \mathbf{c})$ follows a straightforward approach. We firstly encode the
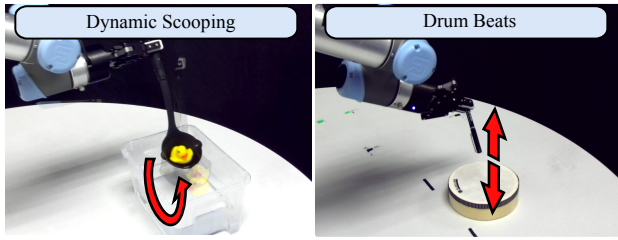
Fig. 4. Real-world tasks in this study: a) "Duck Scooping" task in a water tank, b) "Drum Beats" task by hitting the drum 3 times.

TABLE I
TASK PROPERTIES.

Task	Dis.	HiPrec	Dem.	Act.	Steps
Duck Scoop	0	✓	50	1	∼70
Drum Beats×3	0	✗	150	1	∼70

observation **o** and language condition **c** into the same embedding dimension, and then we utilize the learning objective defined in Equation (3) to train the base policy.

$$\mathcal{L}_{\text{DDPM}} := \mathbb{E}_{\mathbf{o},\mathbf{a},\tau,\mathbf{z}}\left[\|\epsilon(\mathbf{o},\mathbf{a},\tau) - \mathbf{z}\|_2^2\right], \quad (3)$$

The same learning objective is used in finetuning the ControlNet:

$$\mathcal{L} := \mathbb{E}_{\mathbf{o},\mathbf{c},\mathbf{a},\mathbf{a}_{[W_t]},\tau,z}\left[\|\epsilon_\theta(\mathbf{o},\mathbf{c},\mathbf{a}_T,\mathbf{a}_{[W_t]},\tau) - \mathbf{z}\|_2^2\right], \quad (4)$$

where $\mathcal{L}$ is the overall learning objective of the entire diffusion model, $\epsilon_{\boldsymbol{\theta}}(\cdot)$ is the corresponding neural network parameterized by $\boldsymbol{\theta}$. The base policy and the Diff-Control policy are trained end-to-end.

III. REAL-ROBOT TASKS

We conducted a comprehensive evaluation of the ControlNet Policy by comparing it with four baseline methods across two robot tasks. Table I provides a summary of the task properties. The tasks encompassed in our evaluation include: (a) **Duck Scooping** task in a dynamic scenario, (b) **Drum Beats** as a periodic task.

The action of the UR5 robot arm is represented as $\mathbf{a}_{[W_t]}$, where each action is denoted as $\mathbf{a}_i = [x,y,z,r,p,y,g]^T$, where $i \in [1,W]$. It encompasses the position of the end-effector in Cartesian coordinates (x,y,z), the orientations (r,p,y), and the gripper's joint angle g. For all the tasks, the input modalities consist of two modalities: $\mathbf{o} \in \mathbb{R}^{224 \times 224 \times 3}$, corresponds to a RGB image. **c** refers to a language embedding derived from natural language sequences. Task Properties in Table I includes distractors (Dis), number of demonstrations (Dem), number of actions (Act), and whether high-precision is required (HiPrec).

Duck Scooping: Inspired by [17], we explore the interaction between the policy and fluid dynamics. In this task, we equip the robot with a ladle and the robot's objective is to scoop the duck out of the water. As depicted in the bottom right of Figure 4(a), this task presents challenges due to perturbations caused by the entry of the ladle into the water. The flow of water affects the position of the rubber

duck, necessitating the robot to execute precise and cautious movements in order to successfully capture the duck.

Drum Beats: This task is specifically designed for robots to learn periodic motions, a challenging feat due to the unique action representation required [18]. As illustrated in Figure 4(b), the task presents difficulty as the robot must accurately count the number of drum beats and determine when to cease drumming. A total of 150 expert demonstrations were obtained by teleoperating the robot to strike the drum three times in each demonstration.

IV. EVALUATION

The efficacy of the proposed policy is evaluated through two experiments as described in Section III. These experiments aim to address the following questions: (a) Can the Diff-Control policy demonstrate generalization capabilities across diverse tasks? (b) To what extent does the Diff-Control policy outperform the current state-of-the-art methods in terms of overall performance? (c) What are the distinguishing characteristics and benefits of utilizing a stateful policy compared to a non-stateful policy?

We propose the 4 baselines: 1) Image-BC [1], 2) ModAttn [14], 3) BC-Z LSTM, and 4) Diffusion Policy [3]. These baselines all adopt an image-to-action agent framework with varied implementation details. For all experiments, we present the results obtained from the best performing configuration of each baseline method. All the baseline models are reproduced and trained using the collected expert demonstrations for a total of 3,000 epochs. Throughout the training process, checkpoints are saved every 300 epochs. In our analysis, we report the best results achieved from these saved checkpoints for each baseline method.

TABLE II
RESULTS EVALUATION IN FORMS OF SUCCESS RATE (%) AND DURATION (SEC) DURING POLICY EXECUTION

Method	Duck Scoop		Drum Beats	
	Scoop	Duration	Hit × 3	Duration
Image-BC	16%	21.04±1.99	0%	19.50±0.79
ModAttn	12%	20.43±0.16	0%	25.74±0.68
BC-Z LSTM	36%	21.21±1.54	0%	17.53±0.31
Diffusion Policy	76%	21.72±2.17	24%	20.60±1.64
Diff-Control	**84%**	23.12±3.03	**72%**	21.46±1.97

Means±standard errors

A. Duck Scooping Evaluation

In this task, we tested if Diff-Control policy is able to generate consistent actions in a dynamic setting. The success rate and task duration for the given task are presented in Table II. Task duration was recorded as the time interval starting from the policy initiation until the duck was completely removed from the water. The experiment was conducted over 25 trials, with the duck randomly placed in the water for each trial. Among the state-of-the-art methods, the Diff-Control policy achieved a commendable success rate of 84% while performing in this dynamic task. Notably, the Diff-Control policy demonstrated a tendency to successfully
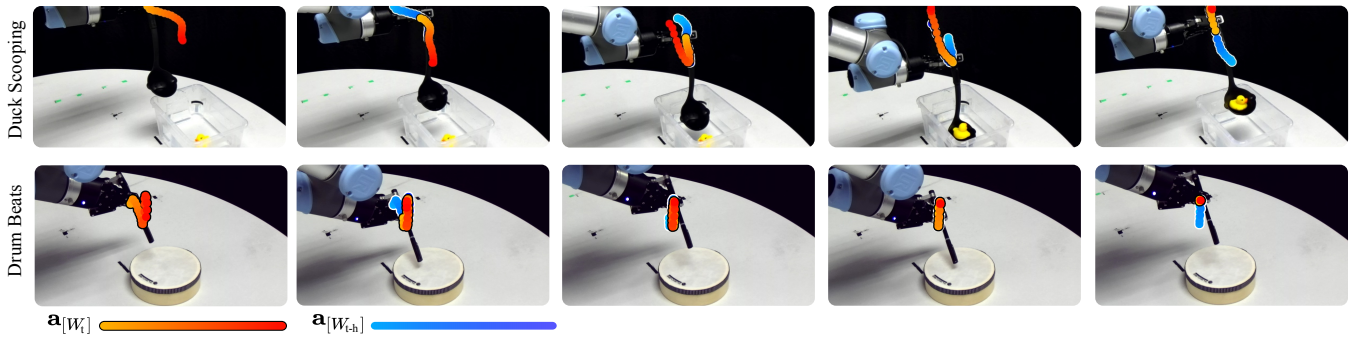
Fig. 5. **Diff-Control for real-world tasks**: The first row shows a successful duck scooping experiment. The second row displays one drum task result. The results are best appreciated with videos on the website: `https://diff-control.github.io/`.
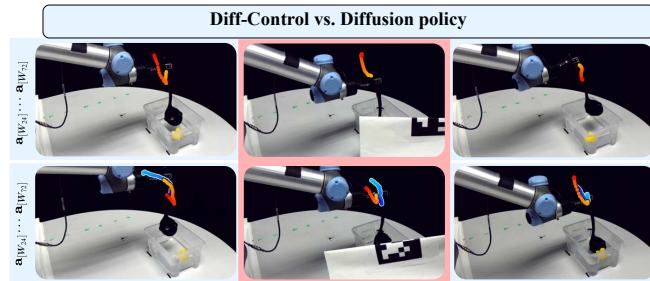

Fig. 6. When visual occlusion is applied, Diff-Control (2nd row) manages to scoop the duck from the water successfully. In contrast, the diffusion policy (1st row) fails. Red shade implies perturbation.

scoop the duck out in a single attempt, reaching a low enough position for accurate scooping. In contrast, the Image-BC and ModAttn failed often as the robot struggled to lower the ladle enough to reach the duck. Diff-Control shows robustness against visual perturbation such as occlusion. In Figure 6 (right), when the view is blocked, the diffusion policy fails immediately. However, Diff-Control can successfully scoop the duck without relying on any visual information because it learns an internal action transition to maintain a stateful behavior.
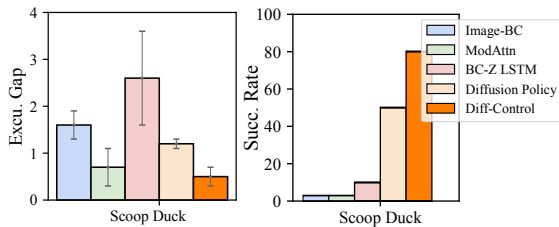

Fig. 7. **Left**: Analysis of the execution gap (measured in cm) for duck scooping tasks during evaluation. **Right**: Assessment of the success rate for each policy when faced with perturbations in the duck scooping task.

Robustness Evaluation: Further evaluation was conducted to assess the consistency of the Diff-Control policy in this task. One way of quantifying action consistency is to measure the execution gap, which is the distance between the tail and head of two consecutive execution windows. In Figure 7(left), the 3D distance of the execution gap is illustrated for each policy network, with Diff-Control displaying the smallest gap. Furthermore, a supplementary
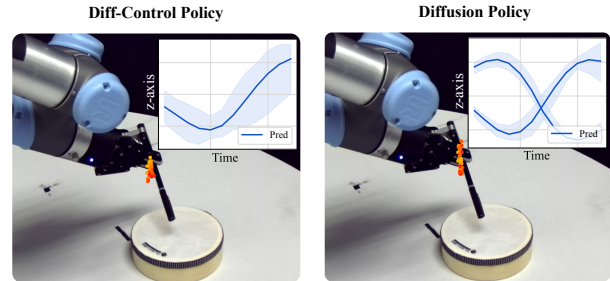

Fig. 8. Generated action with variance from Diff-Control and Diffusion Policy during inference time.

set of experiments was carried out for the duck scooping task under visual occlusion. As depicted in Figure 7, Image-BC and ModAttn only achieved a 0% success rate when faced with perturbation, while BC-Z LSTM succeeded in 1 out of 10 trials in this scenario. Despite the diffusion policy achieving a 50% success rate, Diff-Control demonstrated an 80% success rate, showcasing the benefits of its stateful characteristic.

B. Drum Beat Evaluation

Each policy network was assessed in this experiment to evaluate the impact of statefulness versus non-statefulness on robots learning periodic motions. During testing, success was specifically defined as the robot hitting the drum **exactly three times** and then stopping. The results are presented in Table II, with Diff-Control achieving the highest success rate of 72%. This success rate surpasses the diffusion policy by 48%. The majority of the baseline methods exhibit poor performance (Image-BC, ModAttn, BC-Z LSTM with 0% success rate) due to their inability to accurately predict the direction of actions, such as the end-effector moving upward instead of downward, and the lack of appropriate halting actions. Consequently, The robot is unable to keep track of the number of times it strikes the drum and continues to strike the drum non-stop. While the BC-Z LSTM can accurately count the number of hits, it encounters difficulties in generating reliable actions initially. We visualize one test trial in the last row of Figure 5, where the last plot shows robot stopped after hitting the drum for 3 times.

Furthermore, we compared Diff-Control and the diffusion policy during the inference period, as shown in Figure 8. We

sampled 10 action trajectories from each policy. Interestingly, the diffusion policy, without any prior actions, produced two distributions along the z-axis. This suggests that the policy struggled to determine whether it should descend to strike the drum or ascend after hitting the drum. In contrast, Diff-Control successfully generated actions by striking the drum. This observation shows using **Diff-Control as a stateful policy is beneficial for robot learning periodic behaviors**.

V. CONCLUSION

This study introduces Diff-Control, a stateful action diffusion policy designed for consistent action generation. The study explores the integration of diffusion model with ControlNet for robot action generation, demonstrating how temporal consistency can be enforced to enhance robustness and success rates. Furthermore, our findings underscore the robustness and effectiveness of Diff-Control in managing dynamic and stateful tasks while remaining resilient against perturbations.

REFERENCES

[1] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[2] Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on Robot Learning*, pages 158–168. PMLR, 2022.

[3] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[4] Zhifeng Qian, Mingyu You, Hongjun Zhou, Xuanhui Xu, and Bin He. Robot learning from human demonstrations with inconsistent contexts. *Robotics and Autonomous Systems*, 166:104466, 2023.

[5] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.

[6] Stéphane Ross, Geoffrey Gordon, and Drew 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.

[7] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[8] Suneel Belkhale, Yuchen Cui, and Dorsa Sadigh. Hydra: Hybrid robot actions for imitation learning. *arXiv preprint arXiv:2306.17237*, 2023.

[9] Lucy Xiaoyang Shi, Archit Sharma, Tony Z Zhao, and Chelsea Finn. Waypoint-based imitation learning for robotic manipulation. In *Conference on Robot Learning*, pages 2195–2209. PMLR, 2023.

[10] Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[11] Alexej Klushyn, Richard Kurle, Maximilian Soelch, Botond Cseke, and Patrick van der Smagt. Latent matters: Learning deep state-space models. *Advances in Neural Information Processing Systems*, 34, 2021.

[12] Alina Kloss, Georg Martius, and Jeannette Bohg. How to train your differentiable filter. *Autonomous Robots*, pages 1–18, 2021.

[13] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[14] Yifan Zhou, Shubham Sonawani, Mariano Phielipp, Simon Stepputtis, and Heni Ben Amor. Modularity through attention: Efficient training and transfer of language-conditioned policies for robot manipulation. *arXiv preprint arXiv:2212.04573*, 2022.

[15] Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, 2022.

[16] Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K Wong. Uni-controlnet: All-in-one control to text-to-image diffusion models. *arXiv preprint arXiv:2305.16322*, 2023.

[17] Rika Antonova, Jingyun Yang, Krishna Murthy Jatavallabhula, and Jeannette Bohg. Rethinking optimization with differentiable simulation from a global perspective. In *Conference on Robot Learning*, pages 276–286. PMLR, 2023.

[18] Jingyun Yang, Junwu Zhang, Connor Settle, Akshara Rai, Rika Antonova, and Jeannette Bohg. Learning periodic tasks from human demonstrations. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8658–8665. IEEE, 2022.