Diff-Control: A Stateful Diffusion-based Policy for Imitation Learning

Xiao Liu[1], Yifan Zhou[1], Fabian Weigend[1], Shubham Sonawani[1], Shuhei Ikemoto[2] and Heni Ben Amor[1]

***Abstract*— While imitation learning provides a simple and effective framework for policy learning, acquiring consistent action 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 action representation from a state-space modeling viewpoint. We demonstrate that diffusion-based policies can acquire statefulness 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. Notably, Diff-Control also shows consistent performance in the presence of perturbations, outperforming other state-of-the-art methods that falter under similar conditions. Project page: `https://diff-control.github.io/`**

I. Introduction

In the field of robotics, imitation learning [1] has become a prominent method for programming robots based on expert demonstrations in a data-driven and sample-efficient manner. Within this context, behavioral cloning, in its simplest form, is structured as a supervised regression task aimed at mapping observations to corresponding actions. Previous studies have explored various approaches to learning behavioral cloning policies, such as directly outputting actions via regression models [2] or utilizing implicit policies [3]. Notably, diffusion-based policies [4] 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 [5]. The primary causes of this inconsistency typically stem from the context-rich nature of human demonstrations [6], distribution shift problems [7], and the volatile nature of high-dynamic environments. Previous approaches, such as action chunking [8] and predicting closed-loop action sequences [4], have been proposed to address this issue. Additionally, Hydra [9] and Waypoint-based manipulation [10] modify action representations to ensure consistency. However, these approaches address the problem by altering the action representation without using the actions as is.

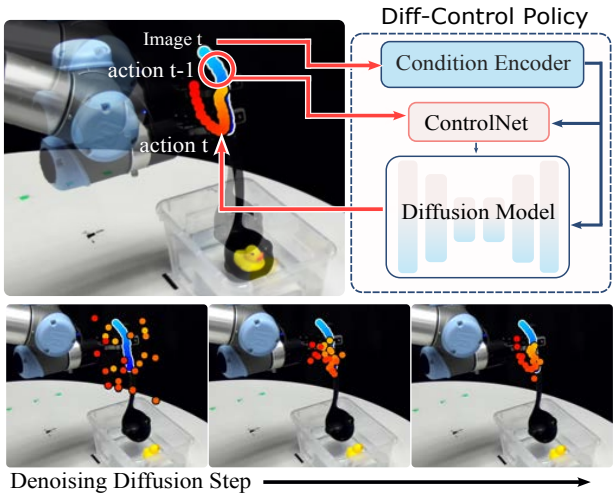

Fig. 1. **Diff-Control Policy** incorporates ControlNet, functioning as a transition model that captures temporal transitions within the action space to ensure action consistency.

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) [11]–[13], the effective learning of a state transition model enables the identification of underlying dynamic patterns. In this paper, we argue that such deep transition 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 [14] 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 10% and 48% improvement in success rate. Diff-Control policy exhibits notable precision and robustness against

[1]Authors are with the School of Computing and Augmented Intelligence, Arizona State University, USA {xliu330, yzhou298, fweigend, sdsonawa, hbenamor}@asu.edu

[2]Author is with Department of Human Intelligence Systems, Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology ikemoto@brain.kyutech.ac.jp

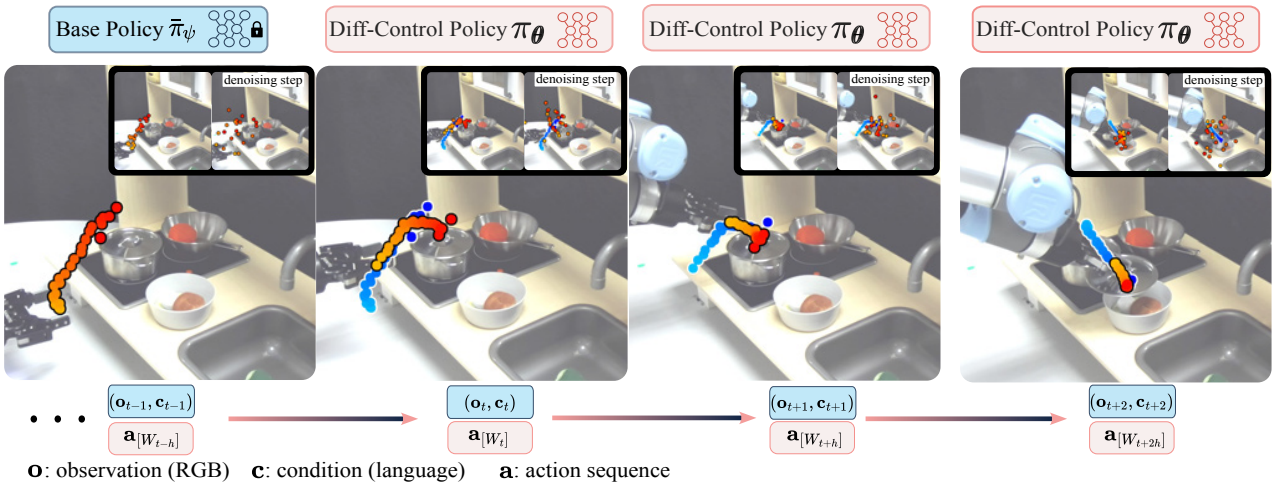

Fig. 2. The **Diff-Control Policy**, denoted as $\pi_{\boldsymbol{\theta}}$, is a stateful diffusion policy. It operates by generating a sequence of actions while incorporating conditioning on previously generated actions. In this example, the Diff-Control policy is depicted executing the "Open Lid" task. For instance, in the second sub-figure, the blue trajectory represents previous action trajectory, denoted as $\mathbf{a}_{[W_t]}$, while the red trajectory displays the newly generated sequence of actions, denoted as $\mathbf{a}_{[W_{t+h}]}$.

perturbations, achieving at least a 30% higher success rate compared to state-of-the-art methods.
- Empirical evidence that Diff-Control Policy can perform on a wide range of tasks including high-precision tasks with at least 31% success rate improvement.

II. Related Work

Diffusion models have gained considerable attention in diverse robot control applications, including tasks such as planning [15], motion prediction [16] and reinforcement learning [17]. Notably, diffusion-based control policies have garnered substantial interest as a robust representation of agent behavior [4], [15]. In the field of imitation learning, these diffusion-based policies have been applied across various tasks, leveraging their capacity to articulate multimodal action distributions. Illustrative applications include goal-directed navigation for mobile robots [18], integration of human behavior in collaborative human-robot tasks [19], and language-guided robot skill learning activities [20]. While these policies demonstrate remarkable scalability in diverse control domains, they are fundamentally stateless, lacking provisions for incorporating memory and prior knowledge into the controller, potentially leading to inconsistent action generation.

To overcome this constraint, integrating a stateful policy [21], [22] becomes imperative, particularly in scenarios where robots are operating in dynamic environments or engaging in tasks over extended horizons. Stateful policies leverage both historical and present states to formulate actions, facilitating adept management of intricate tasks. The application of stateful trajectories has shown benefits in long-horizon skill planning [23], and showcases predictive prowess in evolving environments [24]. A central challenge in this regard is the state representation itself - which features have to be part of the temporal state in order to ensure accurate completion of the task.

In our study, we adopt a deep state-space model (DSSM) approach [11] to address these challenges. DSSMs enable the effective learning of end-to-end models that capture state transitions from observed sequences alone [12], [13]. Differentiable Filters [13] as a subclass of DSSMs has shown the effectiveness of modeling uncertainty with noise profiles in state space modeling. These Differentiable Filters also show good performance on real-world tasks with considerable improvement in state tracking accuracy [25]–[27]. In our work, we leverage a formulation at the intersection of recursive Bayesian filtering and deep learning [26], [28] to construct a state-space model that captures the underlying transition dynamics in both space and time.

III. 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 3: 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 policy [4] 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.

Figure 2 offers an overview of Diff-Control in action for the "Open Lid" task. Within each time window (depicted in

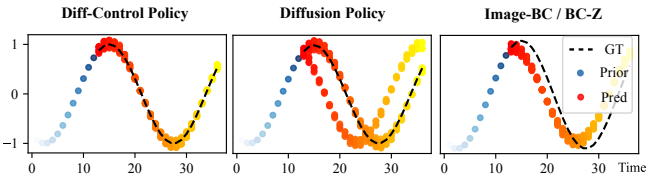

Fig. 3. **Stateful behavior**: at a given state, Diff-Control policy can utilize prior trajectories to approximate the desired function. Diffusion policy [4] learns both modes but fails on generating the correct trajectory cosistently, Image-BC/BC-Z [2] fails to generate the correct trajectory.

red), Diff-Control generates action sequences. When generating subsequent action sequences, it utilizes previous actions as an additional control input, shown in blue. This temporal transition is achieved through Bayesian formulation, effectively bridging the gap between standalone policies and state space modeling.

Subsequently, we will introduce the elements of the Diff-Control algorithm, the underlying network architecture and the training process in detail.

A. Diffusion Model

Diffusion models are generative models that iteratively map Gaussian noise to a target distribution, with the capacity to optionally condition on contextual information [29]. Given $\mathbf{a}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ as the starting point, the diffusion model predicts an output sequence as $\mathbf{a}_{T-1}, \mathbf{a}_{T-2}, \cdots, \mathbf{a}_0$, where each subsequent output serves as a denoised version of the previous output. $\mathbf{a}_0$ is the output after the diffusion process. We use the denoising diffusion model (DDPM) [30] as the backbone. In training, the noise inputs can be generated given a varied noise level: $\mathbf{a}_\tau = \sqrt{\bar{\alpha}_\tau}\mathbf{a} + \sqrt{1-\bar{\alpha}_\tau}\mathbf{z}$, where $\bar{\alpha}_\tau$ is variance schedule, and $\mathbf{z}$ is random noise, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We can train a neural network $\epsilon(\cdot)$ to predict the noise added to the input by minimizing:

$$\mathcal{L}_{\text{DDPM}} := \mathbb{E}_{\mathbf{o},\mathbf{a},\tau,\mathbf{z}}\left[\|\epsilon(\mathbf{o}, \mathbf{a}, \tau) - \mathbf{z}\|_2^2\right], \tag{1}$$

where $(\mathbf{o}, \mathbf{a})$ represents observation and action pairs, τ is the denoising timesteps, $\tau \in [1, T]$. In the sampling step, we interatively run the denoising process:

$$\mathbf{a}_{\tau-1} = \frac{1}{\sqrt{\alpha_\tau}}\left(\mathbf{a}_\tau - \frac{1-\alpha_\tau}{\sqrt{1-\bar{\alpha}_\tau}}\epsilon(\mathbf{o}, \mathbf{a}_\tau, \tau)\right) + \sigma_\tau \mathbf{z}, \tag{2}$$

where σ_τ and α_τ are noise schedule parameters, which has been well-studies in [30], [31].

B. 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 [4], [32], 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 [4], [8], [9], they have not accounted for the statefulness of $\mathbf{a}$. We address the action consistency from a Bayesian perspective

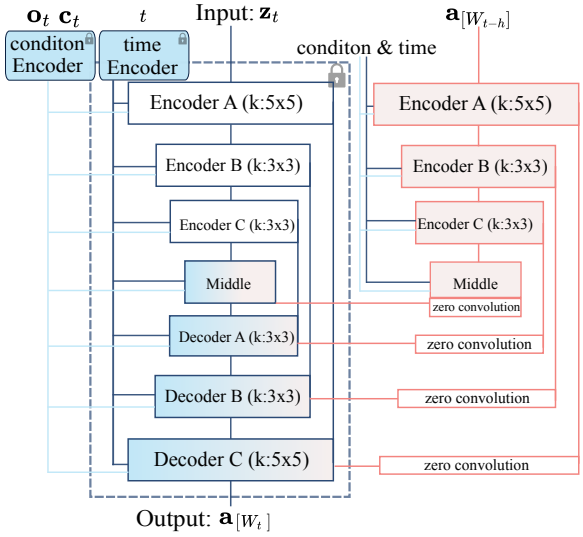

Fig. 4. The Diff-Control Policy is implemented through the utilization of a locked U-net diffusion policy architecture. It replicates the encoder and middle blocks and incorporates zero convolution layers.

by introducing transition in action spaces, our formulation is as follows:

$$\begin{aligned} &p(\mathbf{a}_t|\mathbf{a}_{1:t-1}, \mathbf{o}_{1:t}, \mathbf{c}) \\ &\quad \propto p(\mathbf{o}_t|\mathbf{a}_t, \mathbf{c})\ p(\mathbf{a}_t|\mathbf{a}_{1:t-1}, \mathbf{o}_{1:t-1}, \mathbf{c}). \end{aligned} \tag{3}$$

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

$$\text{bel}(\mathbf{a}_t) = \eta \underbrace{p(\mathbf{o}_t|\mathbf{a}_t, \mathbf{c})}_{\text{observation model}} \prod_{t=1}^{t} \overbrace{p(\mathbf{a}_t|\mathbf{a}_{t-1}, \mathbf{c})}^{\text{transition model}} \text{bel}(\mathbf{a}_{t-1}), \tag{4}$$

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. The transition model describes the laws that govern the evolution of the system dynamics, while the observation model identifies the relationship between the internal state of the system and observed, noisy measurements.

C. 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 $\pi_{\boldsymbol{\theta}}(\mathbf{a}_{[W_t]}|\mathbf{o}, \mathbf{a}_{[W_{t-h}]}, \mathbf{c})$ as shown in Figure 2, 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 $\mathbf{o}$ 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 [4] following the step in Section III-A 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 4). 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 [33]. 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 $\mathbf{a}_{[W_t]}$.

D. 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 observation $\mathbf{o}$ and language condition $\mathbf{c}$ into the same embedding dimension, and then we utilize the learning objective defined in Equation (1) to train the base policy. The same learning objective is used in finetuning the ControlNet:

$$\mathcal{L} := \mathbb{E}_{\mathbf{o},\mathbf{c},\mathbf{a},\mathbf{a}_{[W_t]},\tau,z}\left[\|\epsilon_\theta(\mathbf{o}, \mathbf{c}, \mathbf{a}_T, \mathbf{a}_{[W_t]}, \tau) - \mathbf{z}\|_2^2\right], \quad (5)$$

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.

E. Design Decisions

For all our experiments, we adopt a CNN-based U-net architecture. This choice of a CNN-based backbone has proven to be suitable for a diverse range of tasks, both in simulated and real-world scenarios. In our model architecture, as depicted in Figure 4, the encoder and decoder blocks utilize 1D convolutional layers with varying kernel sizes. To implement the zero convolutional layers for ControlNet, we employ 1D 1×1 convolutional layers with weights initialized to zero. This approach ensures that any potential detrimental noise does not affect the hidden states of the trainable neural network layers during the initial stages of training [14].

We use a window size $W = 24$ as the default predicting horizon for all the experiments, and the execution horizon h between $\mathbf{a}_{[W_{t-h}]}$ and $\mathbf{a}_{[W_t]}$ are 12 steps. We only execute 12 steps ($h = 12$) by default during the experiments in align with [4]. Because according to [4], the execution horizon being too large or too small can cause the drop of the performance.

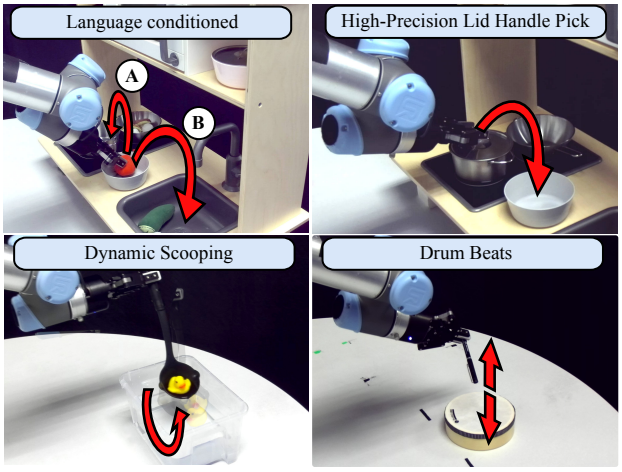

Fig. 5. Real-world tasks in this study: a) language-conditioned pick and place task in kitchen scenario, b) "Open Lid" task with high precision requirement, c) "Duck Scooping" task in a water tank, d) "Drum Beats" task by hitting the drum 3 times.

TABLE I
TASK PROPERTIES

Task	Dis.	HiPrec	Dem.	Act.	Steps
Kitchen Lang.	> 5	$\times$	100	2	$\sim$80
Kitchen Lid	> 5	✓	50	1	$\sim$110
Duck Scoop	0	✓	50	1	$\sim$70
Drum Beats$\times$3	0	$\times$	150	1	$\sim$70

During the implementation of language-conditioned tasks, we observe that diffusion-based policy reaches a certain capacity when learning varied actions utilizing complex CLIP language features [34] as conditions. To address this, a more practical approach is to incorporate a fuse layer and increasing embedding size for visual and language representations, instead of concatenating them directly. This modification can enhance the policy's overall performance for language-conditioned tasks.

IV. REAL-ROBOT TASKS

We conducted a comprehensive evaluation of the ControlNet Policy by comparing it with four baseline methods across five distinct robot tasks. Table I provides a summary of the task properties. The tasks encompassed in our evaluation include: (a) **Language Conditioned** kitchen tasks, (b) **Open Lid** task in the kitchen scene as a high-precision task, (c) **Duck Scooping** task in a dynamic scenario, (d) **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: $\mathbf{o}$ and $\mathbf{c}$. The first modality, $\mathbf{o} \in \mathbb{R}^{224\times224\times3}$, corresponds to a RGB image. The second modality, $\mathbf{c}$, refers to a language embedding derived from natural language sequences. This embedding

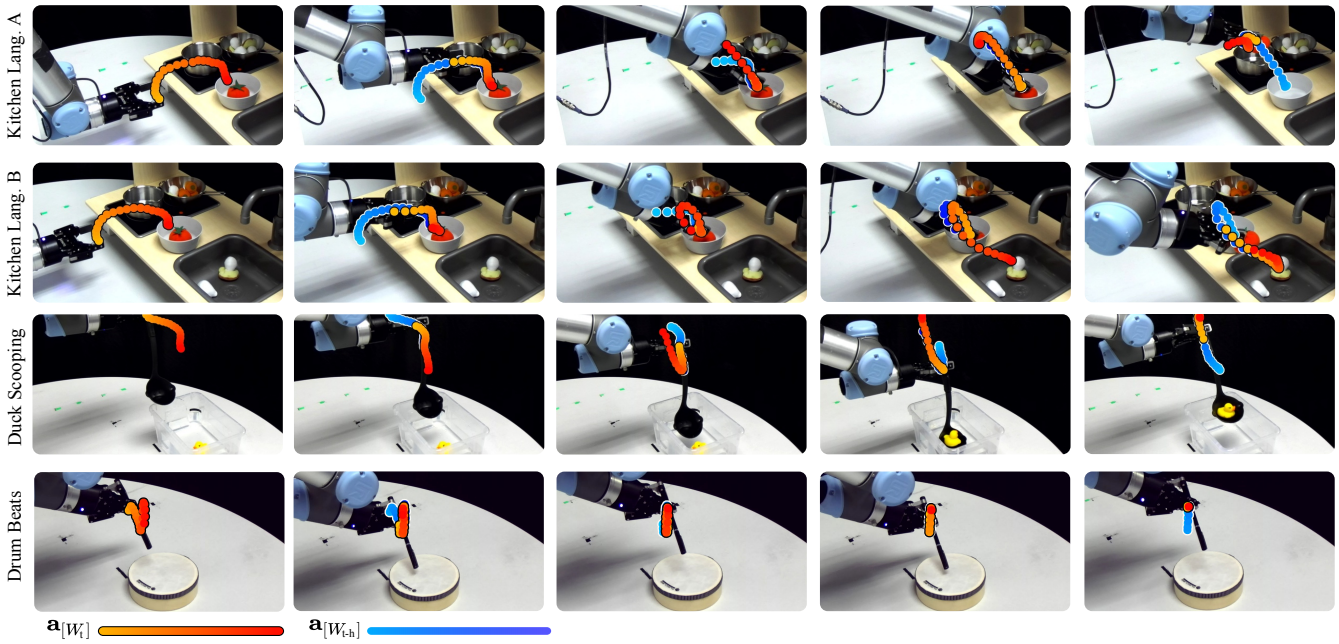

Fig. 6. **Diff-Control for real-world tasks**: The top two rows depict the language-conditioned kitchen task, where the Diff-Control Policy successfully generates actions in alignment with the given language command and consistently executes these actions. The third row shows a successful duck scooping experiment. The last row displays one drum task result. The results are best appreciated with videos on the website: `https://diff-control.github.io/`.

serves as the linguistic input for the robot's understanding and decision-making processes. Table I further arranges the tasks in ascending order of subjective difficulty, providing a summary of task characteristics such as the number of distractors (Dis), number of expert demonstrations (Dem), number of varied actions (Act), and whether high-precision (HiPrec) is required or not. In the following sections, we offer a comprehensive overview of the experimental setups for each task, outline the data collection procedures employed, and the specific challenges encountered within each scenario.

Language Conditioned Kitchen task: This task is designed to resemble several tasks in the kitchen scenario [35], [36]. The robot workspace consists of a scaled-down real-world model kitchen, as illustrated in Figure 5(a). The kitchen environment encompasses various objects, including pots, pans, bowls, and distractor objects resembling plush vegetables. During the data collection process, the distractor objects are randomly positioned. Trained experts are responsible for teleoperating the robot to perform two specific actions within the kitchen environment. The actions involve fetching a tomato and subsequently placing it either in the pot on the stove (A) or in the sink (B) based on the given language instruction.

High-Precision Open Lid: The task includes lifting the lid and subsequently placing it onto a nearby bowl, necessitating precise control. Depicted in Figure 5(b), the lid handle is relatively small, and the lid surface is reflective. To gather data for this task, we obtain 50 expert demonstrations. For each demonstration, we introduce random placements of 5 or more distractor objects, as well as slight shifts in the pot's position and rotations of the lid.

Duck Scooping: Inspired by [37], 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 5(c), 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 [38]. As illustrated in Figure 5(d), 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.

V. Evaluation

The efficacy of the proposed policy is evaluated through four experiments as described in Section IV. 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 baselines are as follows:

1) **Image-BC**: This baseline adopts an image-to-action agent framework, similar to BC-Z [2], it is built upon ResNet-18 backbone and employs FiLM [39] for conditioning using CLIP language features.

TABLE II

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

Method	Kitchen Lang.		Kitchen Lid		Duck Scoop		Drum Beats	
	Pick & Place	Duration	Open Lid	Duration	Scoop	Duration	Hit × 3	Duration
Image-BC	60%	34.48±1.46	12%	35.13±0.50	16%	21.04±1.99	0%	19.50±0.79
ModAttn	28%	36.04±1.51	36%	46.42±4.69	12%	20.43±0.16	0%	25.74±0.68
BC-Z LSTM	64%	35.78±1.69	40%	46.33±7.20	36%	21.21±1.54	0%	17.53±0.31
Diffusion Policy	88%	41.23±2.12	64%	47.94±1.95	76%	21.72±2.17	24%	20.60±1.64
Diff-Control	**92%**	42.51±1.71	**80%**	46.80±1.75	**84%**	23.12±3.03	**72%**	21.46±1.97

Means±standard errors

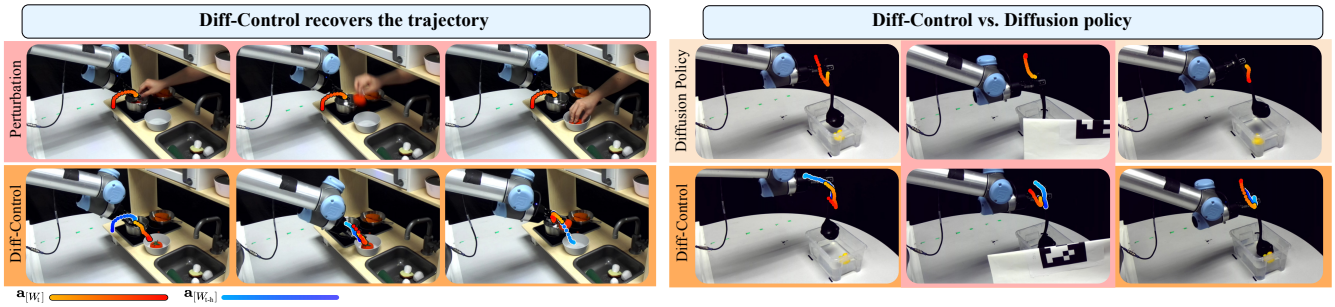

Fig. 7. **Left** shows Diff-Control successfully recovers from a human perturbation, efficiently picking up the tomato.; **Right** shows when visual occlusion is applied, Diff-Control manages to scoop the duck from the water successfully. In contrast, the diffusion policy fails. Red shade implies perturbation.

2) **ModAttn** [32]: This method employs a transformer-style neural network and uses a modular structure to address each sub-aspects of the task via neural attention, it requires a human expert correctly identifies components and subtasks to each task.
3) **BC-Z LSTM**: This baseline represents a stateful policy inspired by the BC-Z architecture. The incorporation of a prior input is achieved by fusing the prior actions and language conditions using MLP and LSTM layers.
4) **Diffusion Policy** [4]: This baseline is a standard diffusion policy.

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. Each experiment was carried out on batch size of 64 on a single NVIDIA Quadro RTX 8000 GPU for roughly 24 hours. For all the tasks, we use the Adamw optimizer with a learning rate of 1e-4.

A. Kitchen Task Evaluation

Language Conditioned: The evaluation results for the language-conditioned kitchen task are presented in Table II. The duration of the task is measured from the initiation of running the policy until the robot successfully opens the gripper and places the tomato either in the sink or on the kitchen stove. We tested each policy for 25 trials. In each of these 25 test trials, we randomized the distractor locations and the pot and bowl placements. Figure 6 shows the Diff-Control policy performing the task with two language-conditions. Among all the baseline methods, Diff-Control achieves the highest success rate with 92%, which is 5%, 64%, and 32% higher than diffusion policy, ModAttn, and Image-BC, respectively. All policies are capable of reaching the correct destination based on the language inputs. Interestingly, we observe that **Diff-Control policy exhibits the ability to recover from perturbations** as shown in Figure 7(left). Diff-Control continues following the language commands whereas ModAttn, and Image-BC failed to do so.

Open Lid: Similar to language-conditioned kitchen task, we randomized the distractor locations and the pot and bowl placements through out the test trials. Diff-Control policy is able to achieve this high-precision task with 80% success rate over 25 trials. The efficacy of the policy is visually illustrated in Figure 2, presenting the diffusion steps involved. In comparison to the baselines, the diffusion policy resulted in a success rate of 64% and the ModAttn policy with 36%. The Image-BC policy yielded the lowest success rate of 0%. It was observed that even a minor offset in the lid position presented substantial challenges for these explicit policies, i.e, Image-BC, ModAttn, and BCZ-LSTM. In contrast, Diff-Control policy demonstrated superior performance and does not show the tendency to overfit on idle actions. This characteristic enabled the gripper to maneuver and explore varied locations for successfully gripping the lid handle.

B. 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

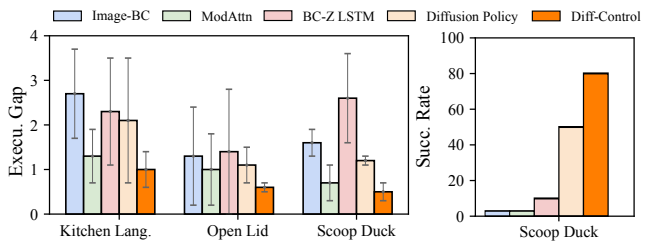

Fig. 8. **Left**: Analysis of the execution gap (measured in cm) for diverse tasks during evaluation. **Right**: Assessment of the success rate for each policy when faced with perturbations in the duck scooping task.

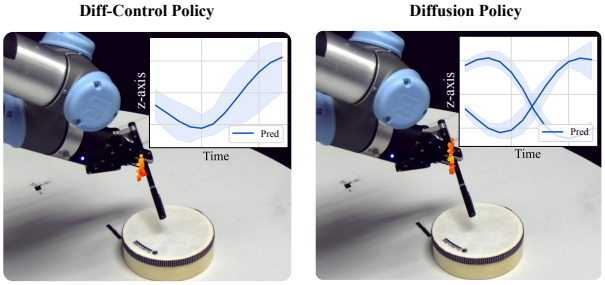

Fig. 9. Generated action with variance from Diff-Control and Diffusion Policy during inference time.

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 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 7 (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.

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 8(left), the 3D distance of the execution gap is illustrated for each policy network, with Diff-Control displaying the smallest gap. Furthermore, a supplementary set of experiments was carried out for the duck scooping task under visual occlusion. As depicted in Figure 8, 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.

C. 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 6, 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 9. 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**.

VI. 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. The efficacy of the proposed policy network is validated across four diverse real-world tasks, each highlighting a unique characteristic of the Diff-Control policy. Furthermore, our findings underscore the robustness and effectiveness of Diff-Control in managing dynamic and stateful tasks while remaining resilient against perturbations.

Limitation: The key limitations that require future consideration are as follows: 1) Diff-Control relies on the assumption that expert demonstrations are optimal, potentially leading to convergence at suboptimal states with insufficient demonstrations; 2) Diff-Control involves fine-tuning on a base policy to acquire action representations, necessitating a two-stage training process. Future work can explore automating the two-stage training process and manipulating different input conditions to ControlNet for other applications, i.e., torque sensors for contact-rich manipulation.

Acknowledgment

This work has received partial support from the National Science Foundation under grants CNS-1932068, IIS-1749783.

REFERENCES

[1] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

[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] C. Jiang, A. Cornman, C. Park, B. Sapp, Y. Zhou, and D. Anguelov. Motiondiffuser: Controllable multi-agent motion prediction using diffusion. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9644–9653, Los Alamitos, CA, USA, jun 2023. IEEE Computer Society.

[17] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193*, 2022.

[18] Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. *arXiv preprint arXiv:2310.07896*, 2023.

[19] Eley Ng, Ziang Liu, and Monroe Kennedy. Diffusion co-policy for synergistic human-robot collaborative tasks. *IEEE Robotics and Automation Letters*, 2023.

[20] Huy Ha, Pete Florence, and Shuran Song. Scaling up and distilling down: Language-guided robot skill acquisition. In *Conference on Robot Learning*, pages 3766–3777. PMLR, 2023.

[21] Atil Iscen, Ken Caluwaerts, Jie Tan, Tingnan Zhang, Erwin Coumans, Vikas Sindhwani, and Vincent Vanhoucke. Policies modulating trajectory generators. In *Conference on Robot Learning*, pages 916–926. PMLR, 2018.

[22] Firas Al-Hafez, Guoping Zhao, Jan Peters, and Davide Tateo. Time-efficient reinforcement learning with stochastic stateful policies. *arXiv preprint arXiv:2311.04082*, 2023.

[23] Utkarsh Mishra, Shangjie Xue, Yongxin Chen, and Danfei Xu. Generative skill chaining: Long-horizon skill planning with diffusion models. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*, 2023.

[24] Marco Monforte, Luna Gava, Massimiliano Iacono, Arren Glover, and Chiara Bartolozzi. Fast trajectory end-point prediction with event cameras for reactive robot control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4035–4043, 2023.

[25] Michelle A Lee, Brent Yi, Roberto Martín-Martín, Silvio Savarese, and Jeannette Bohg. Multimodal sensor fusion with differentiable filters. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10444–10451. IEEE, 2020.

[26] Xiao Liu, Geoffrey Clark, Joseph Campbell, Yifan Zhou, and Heni Ben Amor. Enhancing state estimation in robots: A data-driven approach with differentiable ensemble kalman filters. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1947–1954. IEEE, 2023.

[27] Xiao Liu, Shuhei Ikemoto, Yuhei Yoshimitsu, and Heni Ben Amor. Learning soft robot dynamics using differentiable kalman filters and spatio-temporal embeddings. *arXiv preprint arXiv:2308.09868*, 2023.

[28] Xiao Liu, Yifan Zhou, Shuhei Ikemoto, and Heni Ben Amor. α-mdf: An attention-based multimodal differentiable filter for robot state estimation. In *7th Annual Conference on Robot Learning*, 2023.

[29] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[30] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[31] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[32] 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.

[33] 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.

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[35] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[36] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[37] 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.

[38] 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.

[39] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.