

A Contact Model based on Denoising Diffusion to Learn Variable Impedance Control for Contact-rich Manipulation

Masashi Okada^{†,*}, Mayumi Komatsu[‡] and Tadahiro Taniguchi^{†,*}

Abstract—In this paper, a novel approach is proposed for learning robot control in contact-rich tasks such as wiping, by developing Diffusion Contact Model (DCM). Previous methods of learning such tasks relied on impedance control with time-varying stiffness tuning by performing Bayesian optimization by trial-and-error with robots. The proposed approach aims to reduce the cost of robot operation by predicting the robot contact trajectories from the variable stiffness inputs and using neural models. However, contact dynamics are inherently highly nonlinear, and their simulation requires iterative computations such as convex optimization. Moreover, approximating such computations by using finite-layer neural models is difficult. To overcome these limitations, the proposed DCM used the denoising diffusion models that could simulate the complex dynamics via iterative computations, thus improving the prediction accuracy. Stiffness tuning experiments conducted in simulated and real environments showed that the DCM achieved comparable performance to a conventional robot-based optimization method while reducing the number of robot trials.

I. INTRODUCTION

Learning from demonstration, particularly robot control through the playback of demonstrated trajectories, is a standard technology in the industry. Such controls can be applied to contact-rich tasks using impedance control and careful tuning of its parameters (i.e., time-variant stiffness) to optimize the reproducibility of demonstration (i.e., *task objective*) and safety against unexpected contacts (i.e., *compliance objective*) [1], [2]. In tasks wherein robots have to acquire various skills, tuning must be completed in a short time, ideally in only single demonstration and a few tests. For instance, robotic wiping requires skills to adapt various curved surfaces and trajectories.

In previous literature on contact-rich impedance control, the stiffness was determined using black-box optimization such as Bayesian optimization for the robots to acquire new skills [2]–[8]. Particularly, Ref. [2] utilized multi-objective optimization to find optimal solutions that represent best tradeoff between the *task and compliance objectives*. However, multiple trials have to be performed with real robots (e.g., a few hours of robot operations), which prevent the robots from acquiring a variety of skills.

Robots cannot acquire new skills efficiently without the use of models that accurately simulate contact. If a model

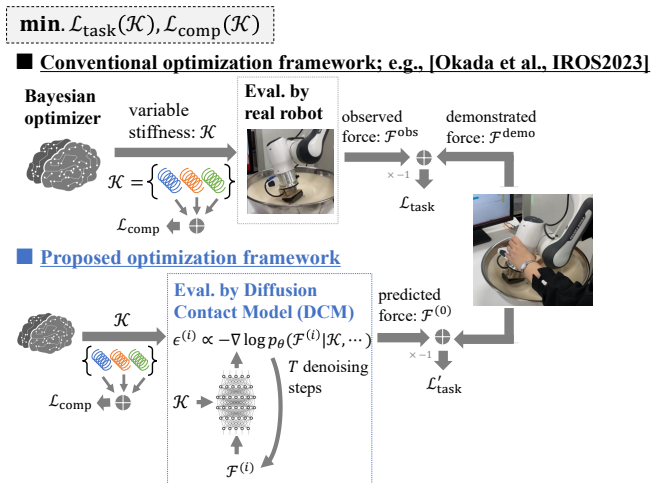


Fig. 1. The concept of learning of variable stiffness \mathcal{K} formulated as a multi-objective optimization of the *task objective* $\mathcal{L}_{\text{task}}$ and *compliance objectives* $\mathcal{L}_{\text{comp}}$. **(Top) Conventional framework (or robot-based optimization)**: the task objective is evaluated using a real robot [2]–[8]. **(Bottom) Proposed framework (or robot-free optimization)**: the task objective is evaluated using the proposed diffusion contact model (DCM) without robot trials.

that precisely simulates robot behavior with contact is developed, model-based stiffness tuning can be performed without robot trials. However, even in state-of-the-art robot simulators such as MuJoCo [9] and PyBullet [10], simulated-to-real (sim2real) gap exists in contact dynamics [11]. In addition, environment models must be constructed for simulating the geometry, friction, elastic properties of the robots and objects; however, constructing such models is difficult.

Contact simulation is difficult due to the high nonlinearity of contact dynamics. Existing robot simulators [9], [10], [12]–[16] formulate contact simulation as a nonlinear complementarity problem (NCP) [17] and solve it via iterative calculations such as convex optimization. Approximating this nonlinearity with neural networks seems a promising approach, and it is also expected to bridge the sim2real gap by exploiting real data. However, deep architectures with extensive parameters are required for the neural modeling of complex dynamics, which need a significant amounts of training data to avoid model overfitting.

Herein, a diffusion contact model (DCM) is proposed to solve the aforementioned problems. DCM employs denoising diffusion models [18], a cornerstone in the recent advancements in *generative AI*, for contact dynamics modeling. Diffusion models were used because they have high nonlinearity due to multi-step denoising. This iterative denoising

[†] Masashi Okada and Tadahiro Taniguchi are with Digital & AI Technology Center, Technology Division, Panasonic Holdings Corporation, Japan.

[‡] Mayumi Komatsu is with Robotics Promotion Office, Manufacturing Innovation Division, Panasonic Holdings Corporation, Japan.

* Tadahiro Taniguchi is also with Ritsumeikan University, College of Information Science and Engineering, Japan.

* okada.masashi001@jp.panasonic.com

can simulate deep networks but with fewer parameters, thus avoiding overfitting. In addition, the denoising process acts as an optimization in a learned objective function [19]. In other words, multi-step denoising and iterative computation for contact simulation (e.g., convex optimization) are similar in terms of optimization, suggesting that DCM can learn the intrinsic process of contact simulation in a data-driven manner.

In this paper, we verify the effectiveness of DCM via simulation and actual experiments of learning robotic wiping as a target task. This task requires a robot to reproduce the demonstrated contact forces between the end-effector and curved surfaces. Figure 1 summarizes the overall concept. This paper follows the framework of a previous study [2] to formulate variable impedance control learning as a multi-objective optimization of the task and compliance objectives. The main contributions of this paper are as follows:

- We introduce DCM for predicting the contact force trajectories of robots from variable stiffness input via iterative denoising based on diffusion models.
- We demonstrate the superiority of DCM compared to a conventional neural model in terms of prediction accuracy of simulated and real wiping trajectories.
- We also show that DCM can perform *robot-free* optimization with comparable performance as that with *robot-based* optimization via robotic wiping experiments.

The remainder of this paper is organized as follows. Section II summarizes related work. Section III briefly reviews the preliminaries, i.e., impedance control, variable impedance control learning as a multi-objective optimization, and denoising diffusion models. Section IV presents DCM, and Sec. V demonstrates its effectiveness. Finally, Sec. VI concludes the paper.

II. RELATED WORK

A. Contact Models in Robotics

Several models have been proposed for estimating the contact forces between objects, such as the linear complementarity problem and the cone complementarity problem (CCP), which are used by off-the-shelf simulators such as MuJoCo [9], Bullet [10], PhysX [12], RaiSim [13], DART [14], Drake [15], and ODE [16]. Each of these methods approximates the NCP of contacts [17], but they commonly require iterative calculations. In particular, MuJoCo reformulates the CCP as a convex optimization problem and solves it using a quadratic programming solver.

B. Robot Behavior Prediction using Neural Models

In the model-based reinforcement learning literature, neural networks are commonly used to predict the future state of robots. Recent studies have shown that the neural models can predict not only robot proprioception [20]–[23] but also high-dimensional vectors such as images [24]–[27]. However, few studies have focused on impedance control or contact force prediction [28], [29]. A neural model was previously used [29] to perform variable impedance control with model

predictive control; however external forces were regarded as actions therein (i.e., input to the neural model). A recurrent neural model [28] was also used to predict the contact force at the next timestep; however, it was not used for planning but for detecting anomalies by comparing the predicted and observed forces.

C. Diffusion Models in Robotics

Neural models in Sec. II-B predict future trajectories autoregressively, thereby accumulating step errors. This issue can be addressed using diffusion models that simultaneously infer all timesteps [30]–[33]. Recently, diffusion models have been commonly used in reinforcement learning to represent policies and value functions [34]–[36].

III. PRELIMINARIES

A. Impedance Control

Impedance control imposes the robot’s dynamics to follow the closed-impedance model:

$$\mathbf{A}\Delta\ddot{\mathbf{x}} + \mathbf{D}\Delta\dot{\mathbf{x}} + \mathbf{K}\Delta\mathbf{x} = \mathbf{F}, \quad (1)$$

$$\Delta\mathbf{x} = \mathbf{x}^{\text{attr}} - \mathbf{x}, \quad (2)$$

where $\mathbf{x} \in \mathbb{R}^6$ is the end-effector pose in task space, $\mathbf{x}^{\text{attr}} \in \mathbb{R}^6$ is the attractor, $\mathbf{F} \in \mathbb{R}^6$ is the external force/torque acting on the end-effector, and the matrices $\mathbf{A} \in \mathbb{R}^{6 \times 6}$, $\mathbf{D} \in \mathbb{R}^{6 \times 6}$, and $\mathbf{K} \in \mathbb{R}^{6 \times 6}$ are the desired Cartesian inertia, damping, and stiffness, respectively. In this context, stiffness \mathbf{K} that exhibits a spring-like behavior enables the robot to follow the desired trajectory of the attractor \mathbf{x}^{attr} while maintaining flexibility in response to unexpected external forces \mathbf{F} . To ensure system compliance, stiffness \mathbf{K} should be maintained low as possible.

B. Formulation to Learn Variable Impedance Control

To achieve safe and satisfactory task performance, variable impedance control is required using appropriately designed stiffness [37], [38]. For instance, let us reproduce a demonstrated trajectory $\tau^{\text{demo}} = \mathbf{x}_{1:H}^{\text{demo}}$, $\mathcal{F}^{\text{demo}} = \mathbf{F}_{1:H}^{\text{demo}}$ with variable stiffness $\mathcal{K} = \mathbf{K}_{1:H}$, where H is the length of the trajectory. The variable stiffness \mathcal{K} affects the safety and reproducibility of the demonstration, and a tradeoff exists the two objectives [1], [2]. Thus, the stiffness learning problem can be formulated as the optimization of two objective functions:

$$\min_{\mathcal{K}} \mathcal{L}_{\text{task}}(\mathcal{K}), \mathcal{L}_{\text{comp}}(\mathcal{K}), \quad (3)$$

where $\mathcal{L}_{\text{task}}$ is the *task objective* and $\mathcal{L}_{\text{comp}}$ is the *compliance objective*. This study defines the objectives as follows:

$$\mathcal{L}_{\text{task}}(\mathcal{K}) := \text{RMSE}(\mathcal{F}^{\text{obs}}(\mathcal{K}), \mathcal{F}^{\text{demo}}), \quad (4)$$

$$\mathcal{L}_{\text{comp}}(\mathcal{K}) := \sum_{t=1}^H |\mathbf{K}_t|, \quad (5)$$

where $\mathcal{F}^{\text{obs}}(\mathcal{K})$ is the observed force on the end-effector. As we focus on the reproduction of demonstrated forces, $\mathcal{L}_{\text{task}}$ is defined as the root-mean-squared error (RMSE) between

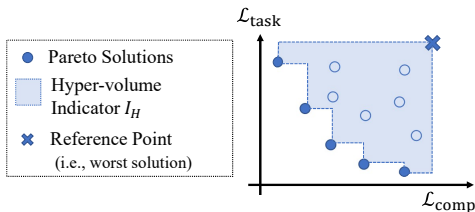


Fig. 2. Illustration of Pareto solutions.

the observed and demonstrated forces. The damping \mathbf{D} and attractor \mathbf{x}^{attr} are determined using the following Eqs. (6), (7) [2]:

$$\mathbf{D} = 2\mathbf{K}^{\frac{1}{2}}, \quad (6)$$

$$\mathbf{x}_t^{\text{attr}} = \mathbf{x}_t^{\text{demo}} + \mathbf{K}_t^{-1}(\mathbf{D}\dot{\mathbf{x}}_t^{\text{demo}} + \mathbf{A}\ddot{\mathbf{x}}_t^{\text{demo}} - \mathbf{F}_t^{\text{demo}}), \quad (7)$$

and are not included in the search space of the optimization problem.

The optimization of the variable stiffness \mathcal{K} for all timesteps is difficult due to the high dimensionality of the search space. The number of dimensions was previously reduced [2] by segmenting a trajectory into several phases and assigning constant responsible stiffness values to each phase. Herein, this segmentation method was used for learning stiffness. The details of the method are beyond the scope of this study and are therefore omitted.

The multi-objective optimization problem in Eq. (3) was solved to obtain several optimal *Pareto solutions*, indicating the best tradeoff between the two objectives. To quantitatively compare the solutions obtained using different methods, the hypervolume indicator I_H was used in the experiments in Sec. V, which is defined by the area composed of Pareto solutions and a reference point. Figure 2 illustrates Pareto solutions and the hypervolume indicator.

C. Diffusion Models

For a dataset \mathcal{D} wherein samples τ are drawn from an unknown data distribution $q(\tau)^1$, diffusion models are used to approximate the distribution using a parameterized generative model $p_\theta(\tau)$. The procedure is completed in two steps: forward diffusion and backward denoising. Forward diffusion generate latents $\tau^{(1:T)}$ by stepwise injection of Gaussian noise into $\tau = \tau^{(0)}$ based on the following transitions: $q(\tau^{(i)}|\tau^{(i-1)}) = \mathcal{N}(\tau^{(i)}; \sqrt{1-\beta_i}\tau^{(i-1)}, \beta_i I)$ with variance schedule $\beta_{1:T}$, yielding the distribution:

$$q(\tau^{(i)}|\tau^{(0)}) = \mathcal{N}(\tau^{(i)}; \sqrt{\bar{\alpha}_i}\tau, (1-\bar{\alpha}_i)I), \quad (8)$$

where $\alpha_i := 1 - \beta_i$, and $\bar{\alpha}_i := \prod_{j=1}^i \alpha_j$. Backward denoising removes the injected Gaussian noise starting from $\tau^{(T)} \sim \mathcal{N}(0, I)$ based on the following denoising transition:

$$p_\theta(\tau^{(i-1)}|\tau^{(i)}) = \mathcal{N}(\tau^{(i-1)}; \mu_\theta(\tau^{(i)}, i), \beta_i I), \quad (9)$$

where

$$\mu_\theta(\tau^{(i)}, i) = \frac{1}{\sqrt{\alpha_i}} \left(\tau^{(i-1)} - \frac{\beta_i}{\sqrt{1-\bar{\alpha}_i}} \epsilon_\theta(\tau^{(i)}, i) \right). \quad (10)$$

¹In this section, τ represents an arbitrary tensor.

The parameterized model $\epsilon_\theta(\tau^{(i)}, i)$ is known as *the score function*, using which the injected Gaussian noise by Eq. (8) can be predicted; it is trained by *the implicit score-matching objective* [19]:

$$\begin{aligned} \mathcal{L}(\theta) &= \mathbb{E}_{i, \tau^{(i)} \sim q(\tau^{(i)}|\tau^{(0)}), \tau^{(0)} \sim \mathcal{D}} \left[\|\epsilon - \epsilon_\theta(\tau^{(i)}, i)\|^2 \right] \\ &= \mathbb{E}_{i, \epsilon \sim \mathcal{N}(0, I), \tau^{(0)}} \left[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_i}\tau^{(0)} + \sqrt{1-\bar{\alpha}_i}\epsilon, i)\|^2 \right], \end{aligned} \quad (11)$$

Specifically, this score function contains the gradients of the learned probability distribution as follows:

$$\epsilon_\theta(\tau, i) \propto -\nabla_\tau \log p_\theta(\tau). \quad (12)$$

IV. METHOD

A. Diffusion Contact Model

In this section, we introduce Diffusion Contact Model (DCM) whose concept is illustrated in Fig. 3.

1) *Formulation*: DCM defines the prediction model as a conditional distribution of $p_\theta(\mathcal{F}|\tau^{\text{demo}}, \tau^{\text{attr}}, \mathcal{K})$, where τ^{demo} is the demonstrated trajectory, $\tau^{\text{attr}} = \mathbf{x}_{1:H}^{\text{attr}}$ is the reference trajectory of the end-effector to be tracked, and \mathcal{K} is the variable stiffness. \mathcal{K} is suggested by a Bayesian optimization algorithm as a solution candidate, and τ^{attr} is calculated from Eq. (7). The demonstrated trajectory τ^{demo} implicitly provides the geometry of the contact surfaces, which is essential for predicting contact forces. The prediction model comprises per-step denoising process $p_\theta(\mathcal{F}^{(i-1)}|\mathcal{F}^{(i)}, (\tau^{\text{demo}}, \tau^{\text{attr}}, \mathcal{K}))$ and score function $\epsilon_\theta(\mathcal{F}^{(i)}, (\tau^{\text{demo}}, \tau^{\text{attr}}, \mathcal{K}), i)$. Differing from Eq. (9), $p_\theta(\mathcal{F}^{(i-1)}|\cdot)$ and $\epsilon_\theta(\cdot)$ are conditioned on $(\tau^{\text{demo}}, \tau^{\text{attr}}, \mathcal{K})$. As shown in Fig. 3(a), this score function is implemented as a neural network that considers the condition variables $(\tau^{\text{demo}}, \tau^{\text{attr}}, \mathcal{K})$ and $\mathcal{F}^{(i-1)}$ as inputs. Figs. 3(b,c) illustrates the input and output of DCM, respectively.

2) *Dataset and Training*: We suppose that conventional Bayesian optimization (the top of Fig. 1) is applied and operated for a certain period using real robots, thus yielding robot operation logs that can be used as a dataset $\mathcal{D} \ni \{(\tau^{\text{demo}}, \tau^{\text{attr}}, \mathcal{K}), \mathcal{F}\}$ to train the score function $\epsilon_\theta(\cdot)$ by optimizing Eq. (11).

3) *Design of the Score Function*: A recurrent neural network (RNN) is used herein to process the time-series input of the score function $\epsilon_\theta(\cdot)$ (Fig. 4(a)), similar to a previous study on the contact force prediction [28] (Fig. 4(b)). However, we introduced Retentive Network (RetNet) [39], a state-of-the-art RNN architecture with performance comparable with that of Transformer [40] while considerably reducing the computational costs. During force prediction, we empirically determined that RetNet exhibited better performance than U-Net [30] and Transformer [36], which are commonly used to implement score functions with time-series input. Regarding the noise schedule defined by $\beta_{1:T}$, we use the linear schedule [18]. The results of preliminary experiment on the design decision are detailed in Appendix. A.

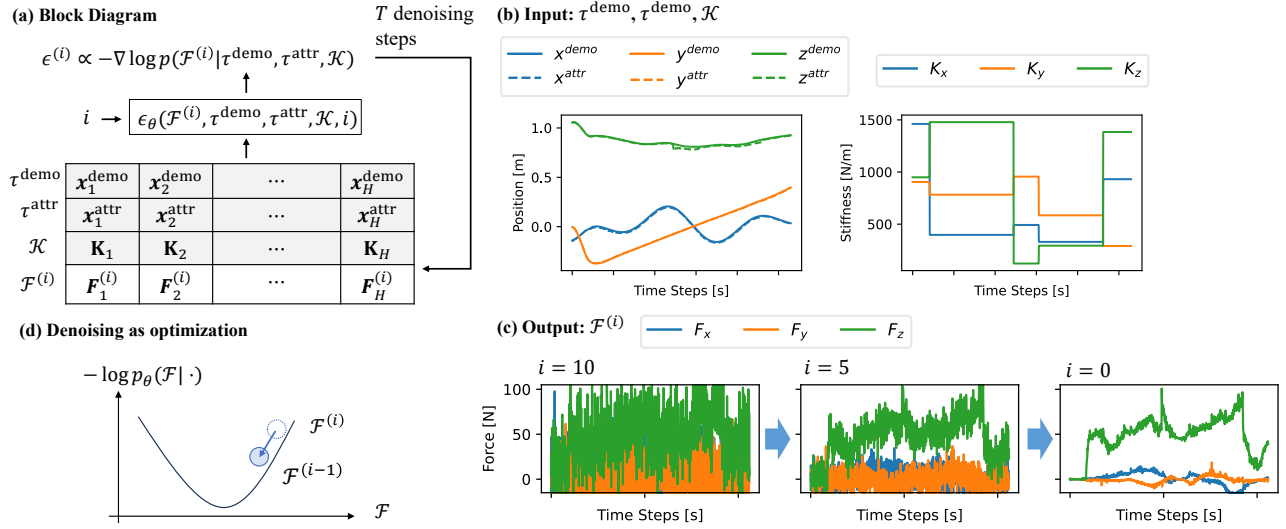


Fig. 3. Conceptual illustration of the diffusion contact model (DCM). (a) **Block-diagram**: DCM iteratively estimates the contact force trajectory \mathcal{F} by T steps of denoising, from the input containing the demonstrated trajectory τ^{demo} , reference trajectory τ^{attr} , and variable stiffness \mathcal{K} . (b/c) **Examples of input/output**, in which the orientation-related sequences are omitted for visibility. (d) **Illustration of iterative denoising process**.

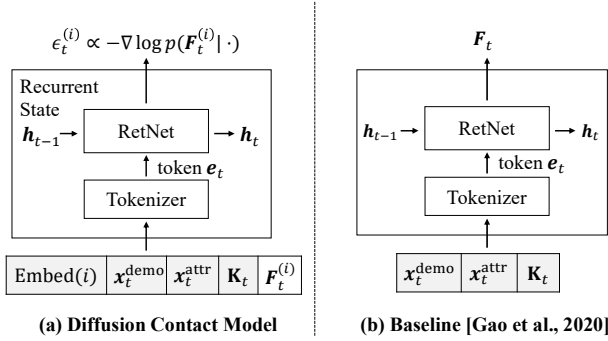


Fig. 4. (a) **Score function implementation** by a state-of-the-art recurrent neural model, RetNet [39]. $\text{Embed}(i)$ denotes the embedding of the denoising step i by the Gaussian Fourier projection [19]. **Tokenizer** is realized by a fully-connected layer, which projects a concatenated input vector to the token e_t . The number of dimensions of e_t was set as 128. (b) **Baseline model** used in Sec. V which predicts forces in a single forward process (i.e., $T = 1$). Bidirectional Gated Recurrent Units (Bi-GRU) were used previously [28]; however RetNet was used here instead.

4) Learning to Simulate Contact via Iterative Denoising:

As can be seen in Eq. (12) and Fig. 3(d), the prediction of \mathcal{F} using DCM can be interpreted as the following optimization:

$$\min_{\mathcal{F}} -\log p_{\theta}(\mathcal{F} | \tau^{\text{demo}}, \tau^{\text{attr}}, \mathcal{K}), \quad (13)$$

where the objective is automatically and implicitly acquired by learning to predict contact forces. A formulation to simulate contact dynamics can be described as the convex optimization as follows [9]:

$$\min_{\ddot{\mathbf{x}}} (\mathbf{\Lambda} \ddot{\mathbf{x}} - \mathbf{F}^{\text{actuated}})^{\top} \mathbf{\Lambda}^{-1} (\mathbf{\Lambda} \ddot{\mathbf{x}} - \mathbf{F}^{\text{actuated}}) + (\mathbf{J} \ddot{\mathbf{x}} - \ddot{\mathbf{x}}^*) \mathbf{R}^{-1} (\mathbf{J} \ddot{\mathbf{x}} - \ddot{\mathbf{x}}^*), \quad (14)$$

where $\ddot{\mathbf{x}}$ is a target variable (i.e., acceleration constrained by contact surfaces), $\ddot{\mathbf{x}}^*$ is reference acceleration, $\mathbf{F}^{\text{actuated}}$ is the forces actuated to robots, \mathbf{J} is the Jacobian, and \mathbf{R} is the

matrix to regularize the constraint: $\mathbf{J} \ddot{\mathbf{x}} = \ddot{\mathbf{x}}^*$. As can be seen in Eqs. (13) and (14), DCM and contact simulation share the same perspective of optimization, suggesting that DCM can acquire the objective $-\log p_{\theta}(\mathcal{K} | \cdot)$ which simulates the objective similar to Eq. (14) in a data-driven manner. By incorporating such an *inductive bias* [41] for contact simulation, we expect that DCM can suppress overfitting and achieve higher prediction accuracy for unseen inputs.

B. Multi-objective Bayesian Optimization by DCM

For a demonstrated trajectory, we tune the variable stiffness \mathcal{K} to acquire the demonstrated skill by solving Eq. (3). As shown in Fig. 1 (bottom), the task objective $\mathcal{L}_{\text{task}}$ is estimated using DCM without conducting trials on real robots. To solve the multi-objective optimization in Eq. (3), we introduce a state-of-the-art Bayesian optimization method proposed in a previous study [42].

V. EXPERIMENTS

We evaluated the effectiveness of the proposed method via simulations and real robot experiments. In the subsequent section, the task and dataset settings are specified in Sec. V-A. Then, the prediction accuracy determined via simulations and real experiments are discussed in Sec. V-B and V-C, respectively. Finally, the effectiveness of DCM in stiffness learning experiments is demonstrated in Sec. V-D.

A. Task and Dataset Preparation

1) *Simulated Environment*: We built a simulated environment for a wiping curved bowl task using MuJoCo [9] and RoboSuite [43] (Fig. 5). In the simulated environment, we collected 1,600 trajectories of spiral motions to train DCM and 30 trajectories of wavelike motions for accuracy evaluation and stiffness learning experiments. The details of the dataset are summarized in the caption of Fig. 5.

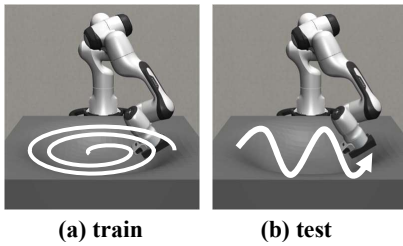


Fig. 5. Simulated wiping task environment and demonstrated trajectories. Demonstrations were performed by manually designed open-loop controllers. **(a) Trajectories for training:** we demonstrated 80 skills of spiral motion with different bowl sizes and directions, and then performed Bayesian optimization with $N = 20$ trials to reproduce each demonstration, which yielded 1,600 trajectories. **(b) Trajectories for test:** we demonstrated a single skill of wavelike motion, and performed Bayesian optimization with $N = 30$ trials.

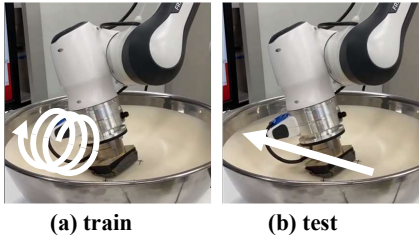


Fig. 6. Real wiping task environment and demonstrated trajectories. Demonstrations were performed by direct teaching. **(a) Trajectories for training:** we demonstrated four skills of spiral motion with different directions, and performed Bayesian optimization with $N = 20$ trials, yielded 80 trajectories. **(b) Trajectories for test:** we demonstrated a single skill of line motion, and performed Bayesian optimization with $N = 20$ trials.

2) *Real Environment:* We also prepared a real environment of the wiping task using Franka Emika Panda, as shown in Fig. 6 for which 80 training trajectories and 20 test trajectories were collected.

B. Prediction Accuracy on Simulation Dataset

We conducted this experiment to answer the following question:

- Does DCM have higher prediction accuracy than conventional neural networks by incorporating iterative calculations?

To this end, training and evaluation were performed with varying diffusion steps T . Here, we regarded $T = 1$ as the baseline method in Ref. [28] and Fig. 4(b), which predicts contact forces in a single forward process of RNN. Two metrics \mathcal{L}_{MSE} and R were used to evaluate the prediction accuracy. \mathcal{L}_{MSE} is the mean-squared error between the predicted and true forces for the test dataset². R is the correlation coefficient between the predicted and true values of the task objective $\mathcal{L}_{\text{task}}$ in Eq. (5), which particularly affects the subsequent Bayesian optimization.

Figure 7 shows the evaluation results of training using the simulated dataset, which demonstrates that DCM outperforms the baseline method (i.e., $T = 1$) on two metrics at a certain step (i.e., $T = 30$). Although the

²We normalized all input and output to DCM to be in the range of $[-1, 1]$, and computed \mathcal{L}_{MSE} using normalized values.

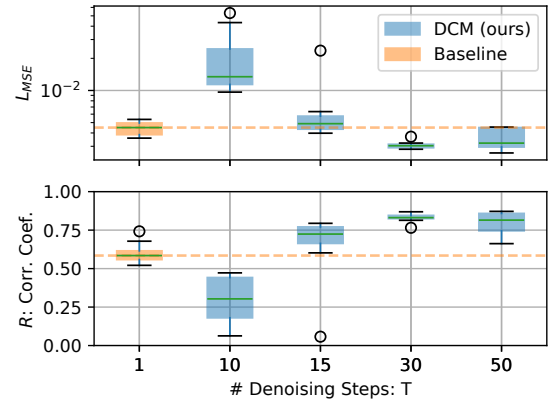


Fig. 7. Prediction accuracies for the simulated dataset with varying diffusion steps T , where $T = 1$ represents the baseline method [28]. Training was conducted eight times for each setting with different random seeds.

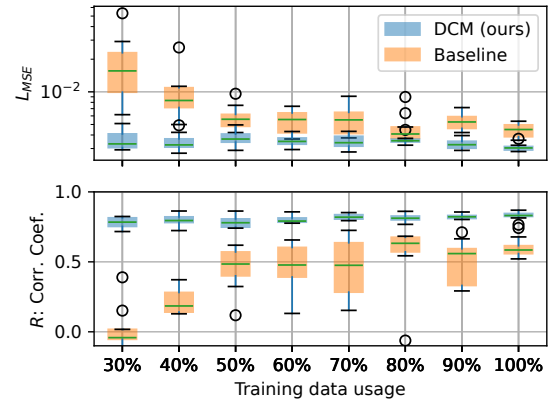


Fig. 8. Prediction accuracies for the simulated dataset with varying usage rate of training dataset, reporting the statistics from eight different seeds for each setting. For DCM, the diffusion steps T was set as 30.

performance improves as T increases, the performance seems to saturate at $T = 30$. Figure 8 shows the effectiveness of DCM from another perspective, in which the prediction accuracy was evaluated using varying usage rates of the training dataset. Although the baseline method and DCM have similar neural architectures, the prediction accuracy of the baseline method degrades rapidly as the usage rate decreases, whereas the performance of DCM remains intact. As discussed in Sec. IV-A.4, this is probably because DCM with iterative calculations acquired the intrinsic procedure of contact dynamics simulation by learning to predict contact forces, thus suppressing overfitting. Figure 9 shows examples of force trajectories predicted by DCM and baseline

C. Prediction Accuracy on Real Dataset

This experiment was conducted to answer the following question:

- Can DCM predict contact forces in real environments?

An issue with real environments is that sufficient training data are rarely available. Therefore, the models pretrained with abundant simulated data were fine-tuned herein.

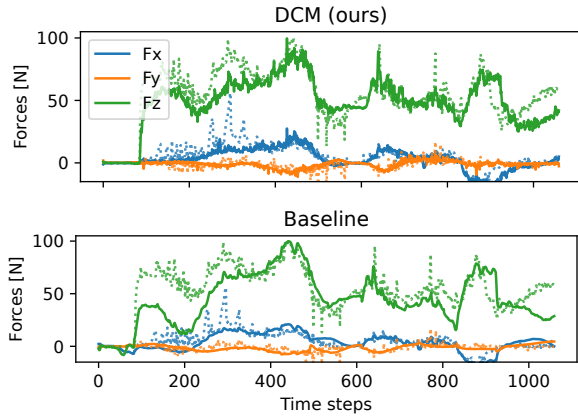


Fig. 9. Examples of the predicted contact force trajectories from *simulated* test data input. Solid and dashed lines represent the predicted and true forces, respectively. F_x and F_y represent the tangential forces (i.e., friction), and F_z represents the normal forces.

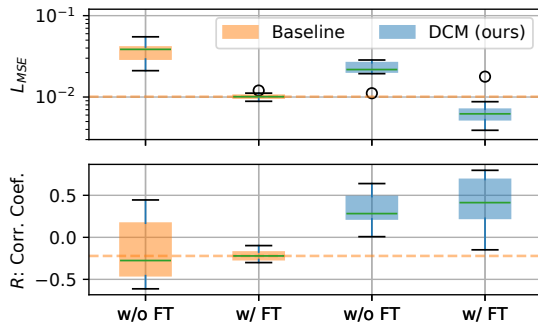


Fig. 10. Prediction accuracies for real dataset. FT denotes *fine-tuning*. Training was conducted eight times for each setting with different random seeds. T was set as 30 for DCM.

DCM and baseline models were trained and tested using the dataset described in Sec. V-A.2 with and without fine-tuning, respectively. Pretrained models were obtained from the previous section. Fig. 10 summarizes the evaluation results, indicating that fine-tuned DCM exhibited the best performance. Although the baseline model did not show any correlation ($R \simeq 0$) between the true and predicted values of $\mathcal{L}_{\text{task}}$, DCM showed a specific positive correlation ($R \simeq 0.5$), suggesting that the learned model can be used for Bayesian optimization. Figure 11 shows the predicted force trajectories by DCM and baseline.

D. Stiffness Learning using DCM

We conducted stiffness learning experiments to verify the following question:

- Can we conduct stiffness learning with DCM while reducing the number of robot trials?

The objective of stiffness learning was to reproduce the test demonstrations in the simulated and real environments. Multi-objective Bayesian optimization (Eq. (3)) was performed, but the task objective $\mathcal{L}_{\text{task}}$ was estimated using DCM and the baseline model. The number of trials N for the robot-free optimization was set as 1,000. The Pareto

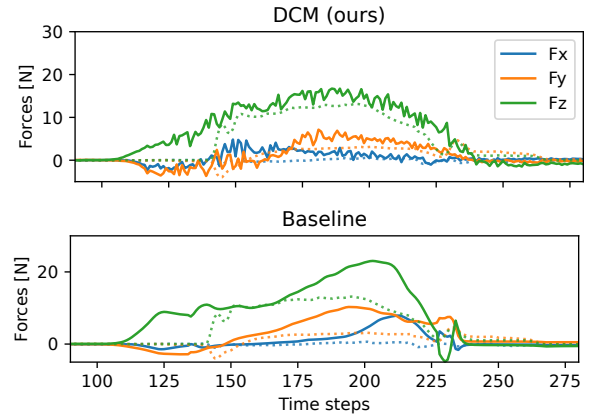


Fig. 11. Examples of predicted contact force trajectories from *real* test data input. Solid and dashed lines represent the predicted and true forces, respectively.

solutions obtained via robot-free optimization were validated by conducting (simulated) robot trials. Then the *genuine* hypervolume indicators I_H were compared. The validity of the *robot-free* optimization results was verified by comparing those with the *robot-based* results (i.e., optimization with N robot trials).

Figure 12 compares the hypervolume indicators I_H in the simulation setting, in which DCM outperformed the baseline method and was comparable to the robot-based optimization with $N = 50$ trials, demonstrating the validity of robot-free optimization performed using DCM. Figure 13 shows the Pareto solutions and realized contact forces.

Figure 14 shows the validity in real settings, in which DCM-based optimization is comparable to the robot-based optimization ($N = 20$) in terms of I_H . The baseline method was not evaluated herein because it failed to predict the task objective $\mathcal{L}_{\text{task}}$ (Fig. 10). In Fig 14, DCM found compliant solutions (i.e., $\mathcal{L}_{\text{comp}} < 80$), but not the task objective-oriented solutions (i.e., $\mathcal{L}_{\text{task}} < 0.025$) unlike robot-based optimization. This suggests that DCM's prediction accuracy in real robot settings must be improved, which is the scope of future studies.

Table I summarizes the optimization cost of the real robot experiment, suggesting that robot-free optimization performed by DCM was more efficient than robot-based (approximately three-fold faster). In the experiment, a single trial was completed in approximately 4 min³, thus robot-based optimization ($N = 20$) was completed in 80 min. In contrast, DCM conducted 1,000 trials virtually in robot-free optimization and found five Pareto solutions in only 5 min. The accuracy of the obtained solutions was validated by conducting five robot tests, which was completed in 20 min. Note that this validation can be eliminated if DCM can achieve sufficient prediction accuracy, leading to further optimization efficiency.

³The trial comprised input verification, rebuilding of a MATLAB control program, robot operation, and environment reset. The control program was rebuilt to apply a new stiffness and was the bottleneck of each trial.

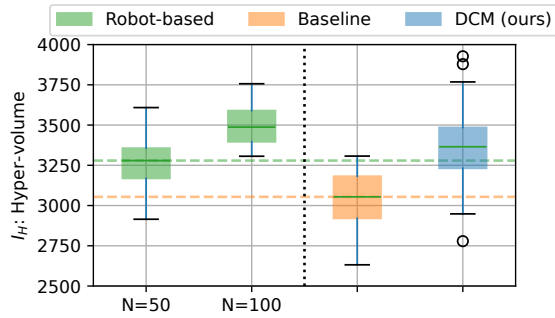


Fig. 12. Hypervolume indicators of the optimization results, where N is the number of simulated robot trials. Experiments were conducted 64 times for each setting using different random seeds.

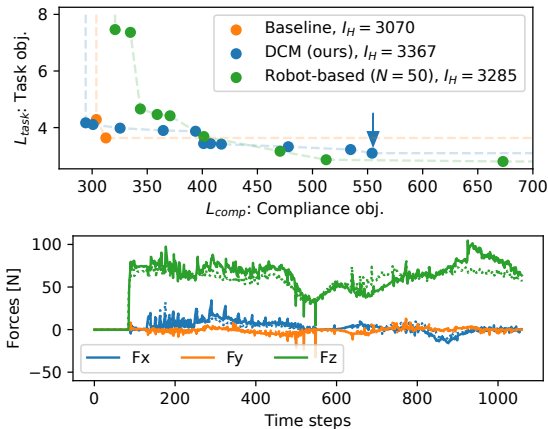


Fig. 13. Simulated stiffness learning results. **(Top)** Pareto solutions of robot-free (using baseline and DCM) and robot-based optimization. **(Bottom)** Contact forces realized by a solution found by DCM (indicated by an arrow in the top part). Solid and dashed lines represent the realized and demonstrated forces, respectively.

VI. CONCLUSION

This paper has proposed DCM for model-based learning of variable impedance control. DCM predicts contact forces from given stiffness and demonstrated trajectories, realizing Bayesian optimization without trials on real robots. To simulate highly nonlinear contact dynamics, diffusion models involving iterative calculations were used. Using these models, inductive bias required for the contact simulation was incorporated. The experiments on prediction accuracy evaluation and stiffness learning demonstrated the effectiveness of DCM in improving the accuracy and efficiency of stiffness

TABLE I

OPTIMIZATION COST IN REAL ROBOT EXPERIMENTS.

	Optimization	Validation	Total Time
robot-based optimization [2]			
# robot trials N	20	-	≈ 80 min.
wall-clock time	≈ 80 min.	-	
robot-free optimization by DCM (ours)			
# robot trials N	1,000 (virtual)	5	≈ 25 min.
wall-clock time	≈ 5 min.	≈ 20 min.	

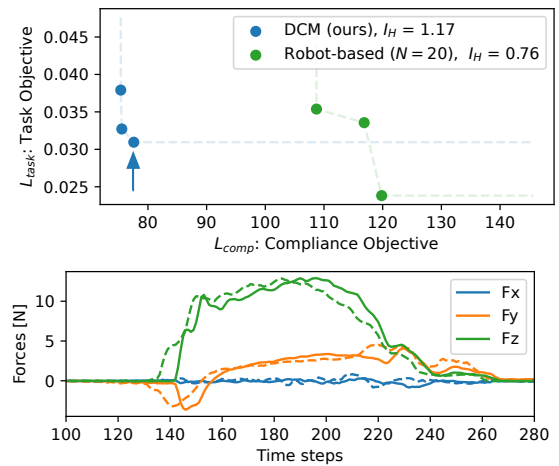


Fig. 14. Real robot stiffness learning results. **(Top)** Pareto solutions of robot-free (DCM) and robot-based optimization. **(Bottom)** Contact forces realized by a solution found by DCM (indicated by an arrow in the top part).

learning.

Moreover, the similarity between DCM and the existing simulators in terms of optimization were discussed. However, DCM is a completely black-box model, and its explainability was not analyzed herein. Explainability analysis and/or the gray-boxing of DCM that incorporates more inductive biases from the analytical contact models will be performed in future research.

Improving the prediction accuracy of DCM particularly in real environments will also be targeted in future research. Recently, substantial volumes of robot demonstrations have been collected to develop *foundational models* for robot control [44], [45]; however, these efforts were mainly focused on pick-and-place operations. If a large dataset of contact-rich tasks can be built, it can be used to enhance the prediction accuracy of DCM. This will allow us to use DCM as a foundational model for force control.

APPENDIX

A. Ablation Study

Figure 15 and 16 show the results of the ablation study for the design decision of the score function $\epsilon_{\theta}(\cdot)$.

REFERENCES

- [1] M. J. Pollayil, F. Angelini, G. Xin, M. Mistry, S. Vijayakumar, A. Bicchi, and M. Garabini, "Choosing stiffness and damping for optimal impedance planning," *IEEE Transactions on Robotics*, 2022.
- [2] M. Okada, M. Komatsu, R. Okumura, and T. Taniguchi, "Learning compliant stiffness by impedance control-aware task segmentation and multi-objective bayesian optimization with priors," in *IROS*, 2023.
- [3] L. Johansmeier, M. Gerchow, and S. Haddadin, "A framework for robot manipulation: Skill formalism, meta learning and adaptive control," in *ICRA*, 2019.
- [4] Z. Wu, W. Lian, C. Wang, M. Li, S. Schaal, and M. Tomizuka, "Prim-LAfD: A framework to learn and adapt primitive-based skills from demonstrations for insertion tasks," *arXiv:2212.00955*, 2022.
- [5] Y. Hu, X. Wu, P. Geng, and Z. Li, "Evolution strategies learning with variable impedance control for grasping under uncertainty," *IEEE Transactions on Industrial Electronics*, 2019.

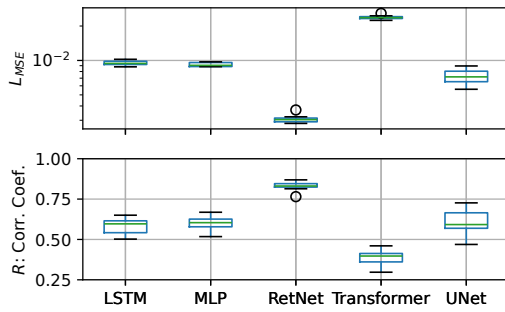


Fig. 15. Ablation study of model architecture of the score function $\epsilon_{\theta}(\cdot)$, in which RetNet exhibited the best performance. In addition to RetNet used in the main experiment, we evaluated the performance of other models such as long-short-time-memory (LSTM), U-Net, Transformer, and multi layer perceptron (MLP); MLP processes each timestep independently.

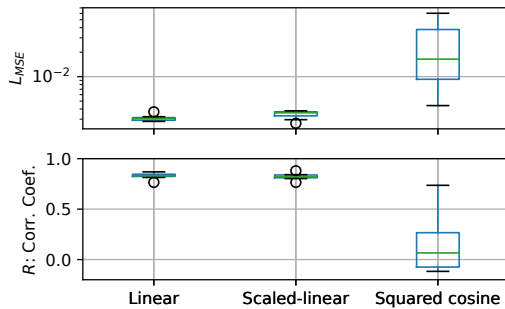


Fig. 16. Ablation study of noise schedules (i.e., variances $\beta_{1:T}$). Previous literature on robotics, wherein diffusion models were used [30], [36] have reported that the squared cosine schedule [46] showed the best performance, but it did not work well in the force prediction task herein.

[6] M. Salehi, G. Vossoughi, M. Vajedi, and M. Brooshaki, "Impedance control and gain tuning of flexible base moving manipulators using pso method," in *ICRA*, IEEE, 2008.

[7] M. M. Fateh and M. M. Zirkohi, "Adaptive impedance control of a hydraulic suspension system using particle swarm optimisation," *Vehicle System Dynamics*, vol. 49, no. 12, pp. 1951–1965, 2011.

[8] V. Azimi, D. Simon, and H. Richter, "Stable robust adaptive impedance control of a prosthetic leg," in *Dynamic Systems and Control Conference*, American Society of Mechanical Engineers, 2015.

[9] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *IROS*, pp. 5026–5033, 2012.

[10] E. Coumans and Y. Bai, "PyBullet, a python module for physics simulation for games, robotics and machine learning," 2016.

[11] Q. L. Lidec, W. Jallet, L. Montaut, I. Laptev, C. Schmid, and J. Carpentier, "Contact models in robotics: a comparative analysis," *arXiv:2304.06372*, 2023.

[12] M. Macklin, K. Storey, M. Lu, P. Terdiman, N. Chentanez, S. Jeschke, and M. Müller, "Small steps in physics simulation," 2019.

[13] J. Hwangbo, J. Lee, and M. Hutter, "Per-contact iteration method for solving contact dynamics," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 895–902, 2018.

[14] J. Lee, M. X. Grey, S. Ha, T. Kunz, et al., "DART: Dynamic animation and robotics toolkit," *The Journal of Open Source Software*, vol. 3, no. 22, p. 500, 2018.

[15] R. Tedrake et al., "Drake: Model-based design and verification for robotics," 2019.

[16] R. Smith et al., "Open dynamics engine, 2008," URL: <http://www.ode.org>, vol. 5, 2005.

[17] V. Acary, M. Brémond, and O. Huber, "On solving contact problems with coulomb friction: formulations and numerical comparisons," *Advanced Topics in Nonsmooth Dynamics: Transactions of the European Network for Nonsmooth Dynamics*, pp. 375–457, 2018.

[18] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *NeurIPS*, vol. 33, pp. 6840–6851, 2020.

[19] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *ICLR*, 2021.

[20] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in *ICRA*, 2018.

[21] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," *NeurIPS*, 2018.

[22] A. Nagabandi, K. Konolige, S. Levine, and V. Kumar, "Deep dynamics models for learning dexterous manipulation," in *CoRL*, PMLR, 2020.

[23] M. Okada and T. Taniguchi, "Variational inference MPC for bayesian model-based reinforcement learning," in *CoRL*, PMLR, 2020.

[24] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," *ICLR*, 2020.

[25] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba, "Mastering atari with discrete world models," in *ICLR*, 2021.

[26] M. Okada, N. Kosaka, and T. Taniguchi, "PlaNet of the bayesians: Reconsidering and improving deep planning network by incorporating bayesian inference," in *IROS*, 2020.

[27] P. Wu, A. Escontrela, D. Hafner, P. Abbeel, and K. Goldberg, "Daydreamer: World models for physical robot learning," in *CoRL*, 2023.

[28] J. Gao, Y. Zhou, and T. Asfour, "Learning compliance adaptation in contact-rich manipulation," *arXiv:2005.00227*, 2020.

[29] A. S. Anand, J. T. Gravdahl, and F. J. Abu-Dakka, "Model-based variable impedance learning control for robotic manipulation," *Robotics and Autonomous Systems*, vol. 170, p. 104531, 2023.

[30] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *ICML*, 2022.

[31] A. Ajay, Y. Du, A. Gupta, J. Tenenbaum, T. Jaakkola, and P. Agrawal, "Is conditional generative modeling all you need for decision-making?," *ICLR*, 2022.

[32] J. Carvalho, A. T. Le, M. Baierl, D. Koert, and J. Peters, "Motion planning diffusion: Learning and planning of robot motions with diffusion models," in *IROS*, IEEE, 2023.

[33] U. A. Mishra and Y. Chen, "Reorientdiff: Diffusion model based reorientation for object manipulation," *arXiv:2303.12700*, 2023.

[34] Z. Wang, J. J. Hunt, and M. Zhou, "Diffusion policies as an expressive policy class for offline reinforcement learning," in *ICLR*, 2023.

[35] P. Hansen-Estruch, I. Kostrikov, M. Janner, J. G. Kuba, and S. Levine, "iDQL: Implicit q-learning as an actor-critic method with diffusion policies," *arXiv:2304.10573*, 2023.

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

[37] F. Ficuciello, L. Villani, and B. Siciliano, "Variable impedance control of redundant manipulators for intuitive human-robot physical interaction," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 850–863, 2015.

[38] K. Kronander and A. Billard, "Stability considerations for variable impedance control," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1298–1305, 2016.

[39] Y. Sun, L. Dong, S. Huang, S. Ma, Y. Xia, J. Xue, J. Wang, and F. Wei, "Retentive network: A successor to transformer for large language models," *arXiv:2307.08621*, 2023.

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NeurIPS*, 2017.

[41] J. Baxter, "A model of inductive bias learning," *Journal of artificial intelligence research*, vol. 12, pp. 149–198, 2000.

[42] Y. Ozaki, Y. Tanigaki, S. Watanabe, and M. Onishi, "Multiobjective tree-structured parzen estimator for computationally expensive optimization problems," in *GECCO*, 2020.

[43] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu, "robosuite: A modular simulation framework and benchmark for robot learning," in *arXiv:2009.12293*, 2020.

[44] A. Brohan, N. Brown, J. Carbajal, et al., "RT-1: Robotics transformer for real-world control at scale," *arXiv:2212.06817*, 2022.

[45] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, et al., "RT-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv:2307.15818*, 2023.

[46] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *ICML*, 2021.