2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Diffusion Stabilizer Policy for Automated Surgical Robot Manipulations

Chonlam Ho[1,*], Jianshu Hu[1,*], Lei Song[2], Hesheng Wang[3], Qi Dou[2], and Yutong Ban[1]

***Abstract*— Intelligent surgical robots have the potential to revolutionize clinical practice by enabling more precise and automated surgical procedures. However, the automation of such robot for surgical tasks remains under-explored compared to recent advancements in solving household manipulation tasks. These successes have been largely driven by (1) advanced models, such as transformers and diffusion models, and (2) large-scale data utilization. Aiming to extend these successes to the domain of surgical robotics, we propose a diffusion-based policy learning framework, called Diffusion Stabilizer Policy (DSP), which enables training with imperfect, perturbed or even failed trajectories. Our approach consists of two stages: first, we train the diffusion stabilizer policy using only clean data. Then, the policy is continuously updated using a mixture of clean and perturbed data, with filtering based on the prediction error on actions. Comprehensive experiments conducted in both simulation and real-world demonstrate the superior performance of our method under different types of perturbations.**

I. INTRODUCTION

Surgical robots have the power of enhancing the capabilities of surgeons by offering greater dexterity and stability in finishing tasks such as suturing [1], [2], tissue manipulation [3], [4], and endoscope control [5], [6]. Furthermore, surgical robots facilitate remote surgeries, paving the way for telemedicine and enabling operation on patients from a distance. The well-known da Vinci Surgical Research Kit (dVRK) system has been widely used in clinical applications. Automation of such surgical robots is an essential step for advancing precision, promoting accessibility and reducing burden of surgeon in medical procedures. However, the automation of such robots for surgical tasks is under-explored compared to recent advancement [7], [8] in solving household tasks using data-driven approaches.

Among data-driven policy learning methods, imitation learning and reinforcement learning have become two predominant approaches. Powered by the ability of modeling multi-modal data distribution, diffusion policy [9] has become a significant branch of work in imitation learning. However, it requires a large amount of data for generalization and its performance highly relies on data quality. Given that imperfect demonstrations with perturbations in real-world data collection are somehow inevitable, properly leveraging perturbed and even failed data could possibly extend the successful application of data scaling in solving household manipulation tasks to the domain of surgical robots.

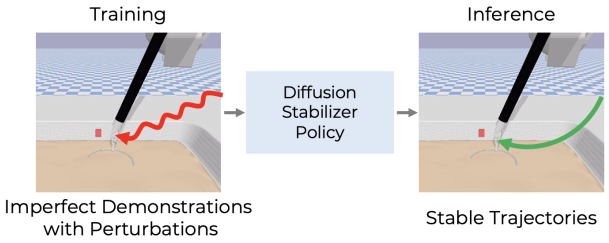

Fig. 1. **Overview.** Our diffuion-based policy learning framework learns a diffusion stabilizer which can filter the perturbed data.

Previous work [10], [11], [12], [13], [14] has explored different imitation learning algorithms with high-quality demonstrations in solving surgical tasks. Due to the inherent property of mimicking the training dataset using imitation learning, perturbed data could even be detrimental to the performance [14]. To avoid the potential issue caused by perturbed data and fully leverage perturbed data, we propose a diffusion-based policy learning framework, called Diffusion Stabilizer Policy (DSP). As shown in Figure 1, this framework allows filtering the imperfect demonstrations and thus making it possible to train with a combination of clean and perturbed data. To specify, we consider dealing with different kinds of imperfect demonstrations in which some of the expert actions are perturbed by some noise. This scenario is common in real-world applications where accidental operation happens or noise of recording device exists during demonstration collection. The key idea in our framework is to first train the diffusion stabilizer policy with only clean data and then apply it as a filter to process the perturbed data. By a comprehensive evaluation on different surgical tasks in SurRoL [15] platform, we demonstrate superior performance of using this diffusion-based policy learning framework when facing both clean data and perturbed data of different types. The main contributions of our work are summarized:

- We propose a diffusion-based policy learning framework for surgical robot manipulations. The framework is able to learn stable manipulation even when perturbations are present in the demonstration.
- We demonstrate the performance of the proposed framework under two types of perturbation: action-level perturbation and trajectory-level perturbation. Our method achieves an overall **31%** performance gain on average success rate under action-level perturbations and **28%**

[1] Chonlam Ho, Jianshu Hu and Yutong Ban are with UM-SJTU Joint Intuitute, Shanghai Jiao Tong University, Shanghai, China {raphaelho2001, hjs1998, yban}@sjtu.edu.cn
[2]Lei Song and Qi Dou are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, HongKong, China qidou@cuhk.edu.hk
[3]Hesheng Wang is with the Department of Automation, Shanghai Jiao Tong Univerisity, Shanghai, China wanghesheng@sjtu.edu.cn
*indicates the authors contributed equally
Corresponding email: yban@sjtu.edu.cn

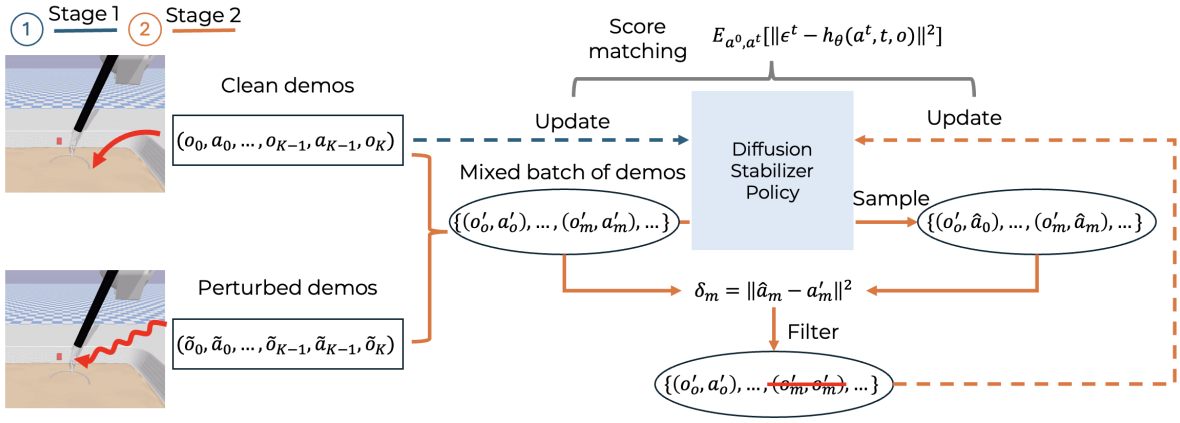

Fig. 2. The overall training framework of **Diffusion Stabilizer Policy.** Our diffusion-based policy learning framework first trains a diffusion stabilizer policy with only clean data. The mixed batch of clean and perturbed data is filtered by the diffusion policy according to the error between predicted actions and the actions from the mixed dataset. The diffusion policy is continuously updated with the filtered data.

performance gain under trajectory-level perturbations respectively.

- We run real-world experiments to demonstrate that the policy trained with imperfect demonstrations can still solve the tasks.

II. Related Work

a) Surgical Simulation Environments: Surgical robots have earned significant attention from researchers and surgeons. Due to the inherent challenges and risks associated with directly conducting real-world experiments, simulation platforms have become an essential tool for evaluating and validating robot learning algorithms before real-world deployment. Different simulation platforms [15], [16], [17] have been developed to provide controlled, reproducible environments containing various surgical robotic tasks. In this paper, we focus specifically on the SurRoL platform [15], which offers a combination of task diversity, realism, and alignment with real-world dVRK setups. Additionally, different baselines [18] varying from reinforcement learning to imitation learning have been evaluated in the environments from SurRoL.

b) Diffusion Policy: Diffusion model is first proposed as a generative model for image generation [19], [20], [21]. By learning the score function of the image data distribution, new images can be generated by first sampling from a normal distribution and then solving a reverse-time SDE [20]. Recently, the ability of diffusion model to learn a data distribution is exploited in imitation learning for solving robotics tasks [9], [22], [23]. A conditional diffusion policy $\pi(a|o)$ is learned from expert demonstrations. In the domain of surgical robots, the application of diffusion policies in surgical tasks is an active research direction [12], [14], [24]. MPD [12] combines diffusion policy and movement primitives [25] to gain gentle manipulation skills. SRT [14] focuses on solving the issue of imprecise joint measurements in real-world setup by introducing a relative action formulation. A comprehensive evaluation of robot learning methods on different surgical tasks are presented in [24].

c) Diffusion Training with Noisy Data: The performance of imitation learning highly relies on the quality of the dataset. A Gaussian policy, commonly exploited in reinforcement learning [26], can resist the noise in the dataset by learning the "mean" of the noisy action. In contrast, the ability of a diffusion model to learn multi-modality in the dataset might hinder its capacity to resist perturbations. This requires diffusion model to be trained on a high quality dataset to achieve satisfactory performance. There is research work on learning with corrupted data [27] or noisy label [28] in training diffusion for image generation. Ambient diffusion [27] requires knowing the corruption matrix in advance for learning with corrupted data, which is often infeasible in the context of diffusion policy. In Label-Noise Robust Diffusion Models [28], the noise in label corresponds to the noise in observations (conditions) to the diffusion policy, which is not compatible with our setting of having noise in the actions. Overall, learning from noisy data remains under-explored, particularly for diffusion policies in surgical environments.

III. Method

To counteract the potential issue of training with perturbed data, we propose a new framework which allows training a diffusion model with a combination of clean and perturbed data, as shown in Figure 2. To specify, this framework trains an diffusion stabilizer policy with only clean data in the first stage. This diffusion stabilizer policy then continues to be updated with a mixture of original clean data and different kinds of perturbed data, which is filtered by the stabilizer policy.

a) Problem Setting: A clean dataset $D = \{\tau_i = (o_{0,\ldots,K}, a_{0,\ldots,K-1}), i = 1, \ldots, N\}$ with N demonstrations τ_i of length K and a perturbed dataset $\tilde{D} = \{\tilde{\tau}_i = (\tilde{o}_{0,\ldots,K}, \tilde{a}_{0,\ldots,K-1}), i = 1, \ldots, \tilde{N}\}$ with $\tilde{N}$ trajectories $\tilde{\tau}_i$ of the same length are generated in the simulated surgical environments. The goal is to learn a policy $\pi(a|o)$ by imitation learning on both clean data D and perturbed data $\tilde{D}$.

We designed two categories of perturbation: action-level perturbation and trajectory-level perturbation. For action-

level perturbation, perturbed actions $\tilde{a}$ are generated by adding noise from a variety of distributions, such as the normal distribution $\epsilon \sim \mathcal{N}(\eta, \sigma)$ to a fixed number of expert/optimal actions a^*: $\tilde{a} = a^* + \epsilon$ within a trajectory. We considered noise distributions including Gaussian, poisson, and uniform noise. This kind of noise corresponds to the noise from the device used for recording the data when collecting demonstrations in a real-world scenario.

Trajectory-level perturbation is used to mimic the situation in which surgeons fail and retry when collecting the demonstrations. For example, in the needle-picking scenario, a perturbed demonstration at the trajectory level may include a trajectory where the surgical robot initially approaches the needle badly, retracts, and reattempts successfully. This type of trajectories departs from the optimal path while still achieving the task. We have defined a set of these trajectory-level perturbation for 6 surgical tasks from the SurRoL environments, including both PSM and Bi-PSM setups, with detailed descriptions provided in the appendix.

b) Overall framework: Imperfect data with perturbations might be detrimental to the performance of the trained diffusion models especially when using limited demonstrations. The main idea of DSP is to first train a diffusion stabilizer policy only on clean dataset and continue to train it with the clean data and the perturbed data after filtering. In the first stage, we train a diffusion policy $\pi_\theta(a|o)$ conditioned on the observations, by learning the conditional score function of the data distribution. The score function is approximated by a diffusion model h parameterized by a Multi-Layer Perceptron (MLP) with parameters θ.

The training of this diffusion policy contains a diffusion process and a denoising process. In the diffusion process, scheduled Gaussian noise with variance β^t is gradually added to the clean action a^0 at diffusion step t:

$$q(a^t|a^{t-1}) = \mathcal{N}(a^t; \sqrt{1-\beta^t}a^{t-1}, \beta^t\mathbf{I}). \tag{1}$$

To avoid confusion, we use the superscript t to indicate the diffusion step, which is different from the subscript k indicating the time step in a trajectory. With the noisy actions a^t, the diffusion model is trained to predict the noise added to it given the diffusion step t and the observation o. The following loss function is used to train the diffusion model:

$$\mathcal{L} = \mathbb{E}_{o,a\sim D}\left[\mathbb{E}_{a^0,a^t}||\epsilon^t - h_\theta(a^t, t, o)||^2\right], \tag{2}$$

where (o, a) are transitions sampled from the clean dataset D, a^t is the noisy action and ϵ^t is the noise added at diffusion step t.

To sample action from the diffusion policy $\pi_\theta(a|o)$, we need to first sample from a Gaussian distribution to get a noisy action a^t, and then repeat the denoising step with the learned score function h_θ:

$$a^{t-1} = \alpha_1(a^t - \alpha_2 h_\theta(a^t, t, o)) + \mathcal{N}(0, \alpha_3\mathbf{I}), \tag{3}$$

where α_1, α_2, α_3 are all constant related to the noise scheduler, only depending on t, used in the diffusion process.

Given that the trained diffusion model already encapsulates knowledge about the underlying data distribution after first stage, it can be utilized to detect abnormal data by calculating the error between the actions predicted by it and those in the perturbed dataset. To filter the perturbed data within a training batch $B = \{(o'_m, a'_m), m = 1, \ldots, M\}$ of M samples from the mixed dataset $D' = D + \tilde{D}$, the trained diffusion model is applied to predict an action $\hat{a}_m$ given each observation o'_m:

$$\hat{a}_m \sim \pi_\theta(\cdot|o'_m), B = \{(o'_m, a'_m), m = 1, \ldots, M\} \sim D' \tag{4}$$

The error δ_m and a threshold γ are used to filter the state action pair (o'_m, a'_m) in the noisy data set.

$$\delta_m = ||\hat{a}_m - a'_m||^2 \tag{5}$$

If the error δ_m is larger than the threshold γ, the transition will not be used for calculating the loss for this batch. The process of sampling data from the mixed dataset, filtering data with current diffusion stabilizer policy and updating it with the filtered data is repeated until the end of training.

c) Implementation Details: We implement our diffusion policy based on CleanDiffuser [29], and adopt the default hyperparameter settings. A four-layer MLP with hidden dimension of 512 is used as the diffusion model. Actions and observations are processed by two different two-layer MLPs respectively into action embeddings and observation conditions with the same dimension of 128. Variance Preserving (VP) continuous-time Stochastic Differential Equation (SDE) [20] is used in the forward process. And the reversed process is solved by a discretized reverse-time VP SDE [19] with 5 denoising steps. The diffusion model is optimized by AdamW [30] with a learning rate of $2 * 10^{-4}$. If not specified, the diffusion policy is trained for 100,000 gradient steps with batch size of 256. We test our method on a computational node with a RTX 4090 GPU and an AMD EPYC 7763 CPU. Specifically, training the diffusion stabilizer policy for 50,000 steps in the first stage requires approximately 30 to 40 minutes. The second stage, which involves online filtering and continuous updating for another 50,000 steps, takes an additional 30 to 40 minutes. This duration varies depending on task complexity—specifically, the dimensionality of the observation and action spaces.

IV. Experiments

In this section, we first validate our proposed method (DSP) in different environments of SurRoL [15] with only clean dataset. Then we show the performance of our method when two types of perturbations are included in the training set. Finally, we present an ablation study of our method under different training settings.

A. Experimental Setup

SurRoL [15] is a well-designed platform to simulate the real-world dVRK system. It contains 10 challenging tasks, which can be classified into 3 categories according to the type of the manipulator: (1) Single-handed patient-sided

TABLE I

PERFORMANCE OF DIFFERENT LEARNING ALGORITHMS IN SURGICAL ENVIRONMENTS

		Reinforcement Learning		Imitation Learning			Demonstration-guided RL					
	Task	SAC[26]	DDPG[31]	BC[32]	SQIL[33]	VINN [34]	DDPGBC [35]	AMP[36]	CoL[37]	AWAC [38]	DEX[18]	**DSP(proposed)**
ECM	Aggregate	0.99(±.03)	0.99(±.02)	**1.00**(±.00)	0.24(±.00)	0.58(±.06)	**1.00**(±.00)	**1.00**(±.01)	**1.00**(±.00)	0.99(±.01)	**1.00**(±.00)	0.97 (±.01)
	ECMReach	**1.00**(±.06)	**1.00**(±.00)	**1.00**(±.00)	0.07(±.04)	0.49(±.10)	**1.00**(±.00)	0.99(±.02)	**1.00**(±.00)	**1.00**(±.00)	**1.00**(±.00)	0.97 (±.02)
	StaticTrack	0.92(±.14)	0.98(±.05)	**1.00**(±.00)	0.43(±.26)	0.56(±.10)	**1.00**(±.00)	0.97(±.03)	**1.00**(±.00)	**1.00**(±.00)	**1.00**(±.00)	0.94 (±.01)
	MisOrient	**1.00**(±.00)	**1.00**(±.00)	**1.00**(±.00)	0.56(±.10)	0.50(±.11)	0.99(±.02)	0.98(±.02)	0.99(±.02)	0.98(±.03)	0.99(±.02)	**1.00** (±.01)
PSM	Aggregate	0.00(±.00)	0.00(±.00)	0.40(±.05)	0.00(±.00)	0.02(±.02)	0.80(±.04)	0.00(±.00)	0.85(±.06)	0.46(±.19)	0.89(±.03)	**0.98** (±.01)
	NeedleReach	**1.00**(±.00)	**1.00**(±.00)	**1.00**(±.00)	0.07(±.09)	0.89(±.06)	**1.00**(±.00)	0.99(±.02)	**1.00**(±.00)	0.94(±.20)	**1.00**(±.00)	**1.00** (±.01)
	GauzeRetrieve	0.00(±.00)	0.00(±.00)	0.07(±.05)	0.00(±.00)	0.01(±.02)	0.63(±.11)	0.00(±.00)	0.71(±.16)	0.43(±.43)	**0.73**(±.12)	**0.73** (±.08)
	NeedlePick	0.00(±.00)	0.00(±.00)	0.21(±.06)	0.00(±.00)	0.02(±.02)	0.91(±.05)	0.00(±.00)	0.96(±.05)	0.26(±.33)	0.94(±.05)	**0.99** (±.02)
	PegTransfer	0.00(±.00)	0.00(±.00)	0.56(±.11)	0.02(±.05)	0.05(±.04)	0.48(±.22)	0.00(±.00)	0.58(±.23)	0.31(±.32)	0.73(±.20)	**0.99** (±.02)
Bi-PSM	Aggregate	0.00(±.00)	0.00(±.00)	0.08(±.04)	0.00(±.00)	0.00(±.00)	0.00(±.00)	0.00(±.00)	0.00(±.00)	0.00(±.00)	0.39(±.11)	**0.83** (±.05)
	NeedleRegrasp	0.00(±.00)	0.00(±.00)	0.09(±.03)	0.01(±.00)	0.01(±.02)	0.05(±.08)	0.00(±.00)	0.04(±.07)	0.00(±.00)	0.63(±.19)	**0.84** (±.07)
	BiPegTransfer	0.00(±.00)	0.00(±.00)	0.09(±.05)	0.00(±.00)	0.00(±.00)	0.00(±.00)	0.00(±.00)	0.01(±.02)	0.00(±.00)	0.18(±.14)	**0.82** (±.08)
	Overall	0.46(±.03)	0.45(±.01)	0.68(±.02)	0.02(±.02)	0.24(±.03)	0.83(±.05)	0.48(±.01)	0.87(±.03)	0.58(±.08)	0.92(±.02)	**0.96** (±.01)

TABLE II

EVALUATION RESULT OF DIFFERENT PERTURBATION FOR DIFFERENT TASKS

		Action-level noise						Trajectory-level noise					
		Gaussian		Poisson		Uniform		Type 1		Type 2		Type 3	
Category	Task	Perturbed	ours	Perturbed	ours	Perturbed	ours	Perturbed	ours	Perturbed	ours	Perturbed	ours
PSM	NeedleReach	0.96 (±.01)	**1.00** (±.01)	0.23 (±.04)	**0.98** (±.02)	0.61 (±.04)	**1.0** (±.00)	0.61 (±.07)	**1.00** (±.00)	**0.99** (±.02)	**0.99** (±.01)	–	–
	NeedlePick	0.38 (±.15)	**0.81** (±.04)	0.69 (±.06)	**0.83** (±.12)	**0.74** (±.09)	**0.74** (±.08)	0.58 (±.10)	**0.79** (±.06)	**0.49** (±.11)	0.43 (±.11)	0.19 (±.05)	**0.60** (±.04)
	GauzeRetrieve	0.67 (±.07)	**0.80** (±.04)	0.30 (±.07)	**0.59** (±.02)	0.71 (±.05)	**0.85** (±.07)	**0.65** (±.07)	0.39 (±.07)	0.26 (±.04)	**0.52** (±.05)	0.05 (±.02)	**0.18** (±.04)
	PegTransfer	**0.83** (±.03)	0.79 (±.06)	0.86 (±.01)	**0.93** (±.02)	0.89 (±.02)	**0.95** (±.05)	0.69 (±.06)	**0.85** (±.02)	**0.83** (±.10)	0.61 (±.09)	0.41 (±.08)	**0.53** (±.10)
Bi-PSM	BiPegTransfer	0.59 (±.06)	**0.74** (±.03)	0.49 (±.05)	**0.73** (±.09)	0.67 (±.08)	**0.71** (±.09)	0.48 (±.12)	**0.57** (±.11)	0.49 (±.08)	**0.57** (±.07)	0.19 (±.06)	**0.70** (±.06)
	NeedleRegrasp	0.40 (±.06)	**0.69** (±.07)	0.35 (±.11)	**0.60** (±.04)	0.62 (±.07)	**0.63** (±.06)	0.76 (±.06)	**0.85** (±.07)	0.63 (±.06)	**0.72** (±.05)	0.43 (±.10)	**0.66** (±.06)

manipulator (**PSM**); (2) Bimanual PSM (**Bi-PSM**); and (3) Endoscopic camera manipulator (**ECM**). The observation space contains the position and orientation of both the end-effector and the object, while Cartesian-space control is applied as actions on the end-effector. We use the same evaluation metrics as used in the baselines [18], which calculate the interquartile mean (IQM) and estimate the 95% confidence interval.

B. Main Results

We start with training a diffusion model using the same number (100 episodes) of expert demonstrations (length of 50) as the baselines [18]. The success rate evaluated in different environments and the aggregated performance are shown in Table I. Our method can maintain comparable performance in relative simple ECM tasks compared to the strong baseline DEX [18], while outperforming all the other baselines in complex tasks. Especially for long-horizon tasks such as BiPegTransfer, which requires the bimanual agent to collaborate on transferring the peg, diffusion policy achieves +355% IQM. As indicated by the aggregated performance for all tasks, leveraging diffusion policy can really push the success rate closer to 100%.

To demonstrate that DSP holds a clear advantage over traditional diffusion-based methods, we use the standard Diffusion Policy [9] as our primary baseline in the perturbation experiments. In the following tables, the columns labeled 'Perturbed' denote the performance of this standard Diffusion Policy trained directly on the mixed datasets without our proposed filtering mechanism.

Building upon this experimental setup, we further evaluate the robustness and generalization of our framework on imperfect demonstration with different types of perturbation. In the experiments, we consider three action-level perturbation types: gaussian, uniform and poisson noise, as described in the Section III. Moreover, we include structured trajectory-level demonstrations that reflect realistic, task-specific suboptimal strategies, such as misapproaches and recovery behaviors. The detailed descriptions for each type of imperfect demonstrations according to their ID are presented in the Appendix. For the threshold, we base on the empirical

TABLE III

EVALUATION RESULT OF ONLINE AND OFFLINE MODE FOR DIFFERENT TASKS

		Action-level noise			Trajectory-level noise		
Category	Task	Perturbed	offline	online	Perturbed	offline	online
PSM	NeedleReach	0.10 (±.03)	0.53 (±.03)	**0.99** (±.02)	0.61 (±.07)	0.65 (±.06)	**1.00** (±.00)
	NeedlePick	0.61 (±.07)	**0.93** (±.04)	**0.93** (±.04)	0.58 (±.10)	0.46 (±.13)	**0.79** (±.06)
	GauzeRetrieve	**0.73** (±.08)	**0.73** (±.08)	**0.73** (±.08)	**0.65** (±.07)	0.59 (±.06)	0.39 (±.07)
	PegTransfer	**0.89** (±.04)	0.86 (±.07)	0.88 (±.04)	**0.83** (±.10)	0.81 (±.05)	0.61 (±.09)
Bi-PSM	BiPegTransfer	0.52 (±.04)	0.50 (±.09)	**0.57** (±.03)	0.48 (±.12)	**0.77** (±.07)	0.57 (±.11)
	NeedleRegrasp	0.49 (±.04)	**0.69** (±.05)	0.54 (±.08)	0.76 (±.06)	**0.88** (±.05)	0.85 (±.07)
Aggregate		0.58 (±.03)	0.71 (±.03)	**0.80** (±.02)	0.65 (±0.03)	**0.71** (±0.03)	0.69 (±0.03)

TABLE IV

EFFECTS OF DIFFERENT DEMONSTRATION AMOUNTS

Method	Number of episodes in demonstrations				
	10 epi.	25 epi.	50 epi.	75 epi.	100 epi.
BC [32]	0.00 (±.00)	0.05 (±.02)	0.15 (±.03)	0.19 (±.03)	0.22 (±.03)
SQIL [33]	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)
VINN [34]	0.03 (±.03)	0.02 (±.03)	0.02 (±.02)	0.01 (±.02)	0.01 (±.01)
DDPGBC [35]	0.00 (±.03)	0.20 (±.11)	0.35 (±.05)	0.47 (±.10)	0.45 (±.08)
AMP [36]	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)
CoL [37]	0.18 (±.07)	0.47 (±.09)	0.46 (±.09)	0.49 (±.05)	0.43 (±.10)
AWAC [38]	0.00 (±.05)	0.05 (±.14)	0.07 (±.18)	0.40 (±.15)	0.39 (±.13)
DEX [18]	**0.24** (±.02)	0.50 (±.09)	0.68 (±.07)	0.78 (±.07)	0.80 (±.06)
DSP (Proposed)	0.15 (±.02)	**0.67** (±.03)	**0.84** (±.02)	**0.91** (±.02)	**0.92** (±.02)

TABLE V

EVALUATION WITH DIFFERENT PERTURB RATIO AND TOTAL EPISODE AMOUNT FOR DIFFERENT TASKS

	Action-level Perturb						Trajectory-level Perturb					
Env	$(N, \tilde{N}) = (15, 15)$		$(N, \tilde{N}) = (25, 25)$		$(N, \tilde{N}) = (15, 35)$		$(N, \tilde{N}) = (15, 15)$		$(N, \tilde{N}) = (25, 25)$		$(N, \tilde{N}) = (15, 35)$	
models	perturbed	ours	perturbed	ours	perturbed	ours	perturbed	ours	perturbed	ours	perturbed	ours
NeedleReach	0.03 (±0.02)	**0.96** (±0.02)	0.10 (±0.03)	**0.99** (±0.02)	0.41 (±0.12)	**0.99** (±0.01)	0.43 (±0.08)	**0.99** (±0.02)	0.61 (±0.07)	**1.00** (±0.00)	0.27 (±0.06)	**0.94** (±0.01)
GauzeRetrieve	0.40 (±0.02)	**0.73** (±0.03)	**0.73** (±0.08)	**0.73** (±0.08)	0.51 (±0.02)	**0.76** (±0.07)	**0.31** (±0.07)	0.21 (±0.04)	**0.65** (±0.07)	0.39 (±0.07)	**0.23** (±0.04)	0.13 (±0.03)
NeedlePick	0.06 (±0.07)	**0.55** (±0.04)	0.61 (±0.07)	**0.93** (±0.04)	0.55 (±0.09)	**0.96** (±0.02)	0.46 (±0.13)	**0.63** (±0.08)	0.58 (±0.10)	**0.79** (±0.06)	0.48 (±0.08)	**0.55** (±0.11)
PegTransfer	**0.45** (±0.08)	0.37 (±0.03)	**0.89** (±0.04)	0.88 (±0.04)	**0.87** (±0.04)	0.81 (±0.06)	**0.63** (±0.10)	0.36 (±0.11)	**0.83** (±0.10)	0.61 (±0.09)	**0.77** (±0.05)	0.37 (±0.07)
NeedleRegrasp	0.39 (±0.05)	**0.67** (±0.04)	0.49 (±0.04)	**0.54** (±0.08)	0.27 (±0.04)	**0.35** (±0.02)	0.49 (±0.05)	**0.51** (±0.06)	0.76 (±0.06)	**0.85** (±0.07)	**0.71** (±0.04)	0.61 (±0.08)
BiPegTransfer	0.41 (±0.10)	**0.64** (±0.09)	0.52 (±0.04)	**0.57** (±0.03)	0.59 (±0.02)	**0.75** (±0.07)	0.53 (±0.07)	**0.54** (±0.08)	0.48 (±0.12)	**0.57** (±0.11)	**0.73** (±0.09)	0.43 (±0.07)
Aggregate	0.32 (±0.03)	**0.65** (±0.02)	0.58 (±0.03)	**0.80** (±0.02)	0.52 (±0.03)	**0.81** (±0.03)	0.48 (±0.03)	**0.51** (±0.03)	0.65 (±0.03)	**0.69** (±0.03)	**0.56** (±0.03)	0.49 (±0.04)

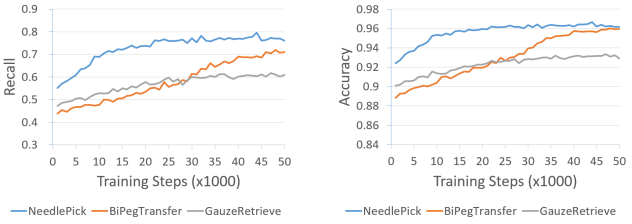

Fig. 3. The perturbed samples generated during data collection and their filtering results are recorded. The left figure illustrates the recall of the predictions, representing the percentage of correctly identified perturbed data. The right figure depicts the accuracy, indicating the percentage of correctly classified samples across the entire dataset.

mean $\hat{\mu}$ and the empirical variance $\hat{\sigma}$:

$$\hat{\mu} = \frac{1}{M}\sum_{m=1}^{M}\delta_m, \hat{\sigma}^2 = \frac{1}{M-1}\sum_{m=1}^{M}(\delta_m - \hat{\mu})^2, \quad (6)$$

Two thresholds: $\hat{\mu}$ and $\hat{\mu} - \hat{\sigma}$, are used in the training with action-level perturbation and trajectory-level perturbation, respectively. The results, summarized in Table II, indicate that our filtering mechanism consistently improves policy performance across different noise conditions and imperfect demonstration types. Notably, our method effectively identifies and suppresses both stochastic perturbations and systematic biases, leading to higher task success rates compared to naive inclusion of perturbed data.

C. Ablation Study and Analysis

In this section, we design experiments to answer the following questions: 1) Does our method successfully filter the perturbed data? 2) Does our method work with different numbers of clean and perturbed data? 3) What are the results of using different thresholds for filtering?

a) Does our method successfully filter the perturbed data?: We answer the first question by calculating 1) **the recall:** the percentage of perturbed samples which are correctly filtered by our diffusion stabilizer policy; 2) **the accuracy:** the percentage of samples which are correctly classified by our diffusion stabilizer policy along the training in the second stage. The experiments are conducted on a subset of tasks, including 3 representative tasks: NeedlePick, GauzeRetrieve, and BiPegTransfer. As shown in Figure 3, both the recall and accuracy keep increasing along the training and the diffusion stabilizer policy can achieve a high accuracy at the end of training. Furthermore, two variations of the filtering methods under our framework, marked as "online" and "offline", are also tested. In the "offline" mode, the entire perturbed dataset is evaluated and filtered a single time using the frozen model weights obtained at the end of the first stage. Conversely, the "online" mode continuously evaluates and filters the mixed batch during the second training stage using the actively updating model weights. This allows dynamically adjusting its filtering criteria as the approximation of the data distribution improves, leading a better classification of borderline samples. The results are shown in Table III, demonstrating the effectiveness of our method in detecting perturbed data and highlighting the advantages of using an online mode over an offline mode.

b) Does our method work with different numbers of clean and perturbed data?: First, to present the effect of varying numbers of clean demonstrations, a comprehensive evaluation is conducted on the PSM and Bi-PSM tasks.

As shown in Table IV, the performance of the proposed DSP method begins to saturate around 75 to 100 episodes, indicating that 100 episodes are generally sufficient for the diffusion policy to capture the task distribution. The stringent success rate requirements of surgical robotics and complexity of real surgical tasks still highlight the necessity fully leveraging limited expert demonstrations. Therefore, effectively utilizing all available data, including perturbed trajectories, remains critical.

To demonstrate the effectiveness of our framework with different constructions of the training set, we construct three mixed datasets containing N clean demonstrations and $\tilde{N}$ perturbed episodes, with $(N, \tilde{N}) \in (15, 15), (25, 25), (15, 35)$. This setup evaluates two aspects of dataset construction, as shown in Table V. First, we compare the performance of our method under the same ratio of

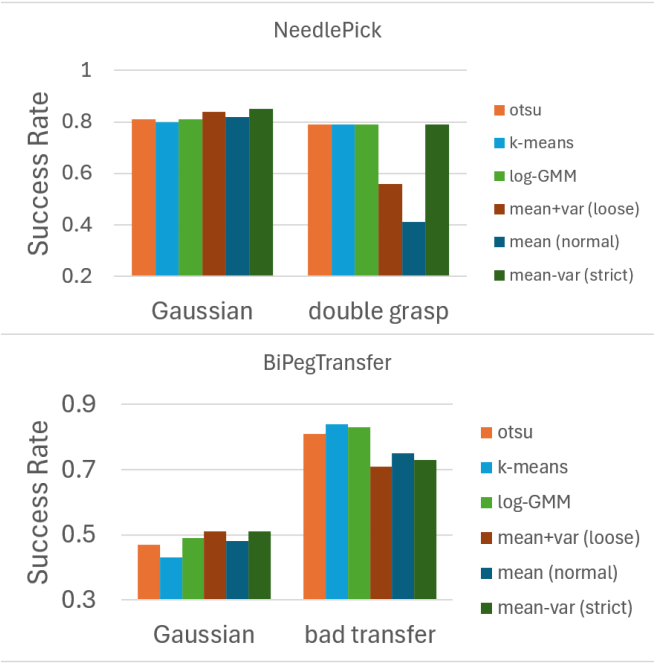

Fig. 4. Comparison of six thresholding methods across two surgical tasks (NeedlePick and BiPegTransfer) under two distinct noise configurations (Gaussian step-perturbed and imperfect demonstrations). Results demonstrate that our method maintains robust performance across different threshold selections in both task environments and noise conditions.

clean and perturbed data, i.e., $(N, \tilde{N}) = (25, 25)$ vs. $(15, 15)$, while scaling up the dataset size. With larger datasets, the diffusion policy can learn a better representation of the data distribution from the additional clean data, thereby reducing the negative impact of perturbed data. Second, we analyze the effect of varying the ratio of clean to perturbed data under the same dataset size, i.e., $(25, 25)$ vs. $(15, 35)$. It is worth noting that in the last two columns of Table V for Trajectory-level Perturbation (e.g., $(N, \tilde{N}) = (15, 35)$), the performance of our method is unexpectedly worse than the baseline for certain tasks like PegTransfer and GauzeRetrieve. This phenomenon occurs because trajectory-level perturbations often contain multi-modal recovery behaviors. Our strict filtering mechanism may inadvertently discard these complex but valid boundary samples, excessively reducing the effective dataset size. Consequently, the diffusion model struggles to generalize, whereas the standard diffusion baseline benefits from the sheer volume of the mixed data despite its noise.

c) What are the results of using different thresholds for filtering?: As stated in Section III, we use a threshold γ to filter the perturbed data within a batch. To further investigate the sensitivity of our method to threshold selection, we introduce several alternatives, which includes a stricter alternative $\hat{\mu} - \hat{\sigma}$ and a loose threshold $\hat{\mu} + \hat{\sigma}$:

$$\begin{aligned} \gamma_{strict} &= \hat{\mu} - \hat{\sigma}, \\ \gamma_{loose} &= \hat{\mu} + \hat{\sigma}, \end{aligned} \tag{7}$$

where $\hat{\sigma}^2$ is the empirical variance of the error. These two thresholds are designed to exclude or accept more of samples in each batch respectively. Beyond pre-defined thresholds, we also explore adaptive thresholding techniques including Otsu's method [39], Gaussian Mixture Models (GMM) [40], and K-means clustering [41]. Given that the error distribution is heavily concentrated near zero, we apply logarithmic scaling to the errors before computing adaptive thresholds, followed by exponential transformation to revert to the original scale. The results are shown in Figure 4.

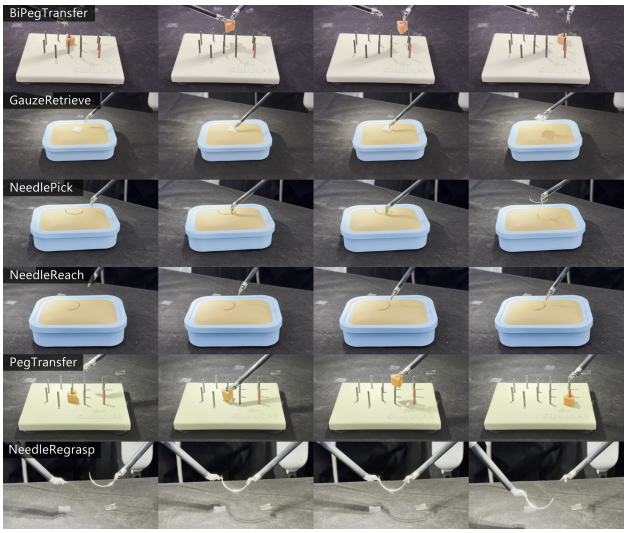

Fig. 5. Keyframe sequences of task executions on the real robotic platform. Each row corresponds to a surgical task, illustrating its complete workflow. Each column represents the execution state of different tasks at a similar phase (e.g., approach, grasp, transfer, placement).

D. Real-World Demonstration

To validate the practical applicability and sim-to-real transfer capability of our framework, we successfully deployed the policies trained in simulation onto an actual robotic surgical platform, conducting functional validation for all six surgical tasks outlined in our study. As visually documented in Figure 5, these real-world demonstrations capture the complete execution workflow of each task, showcasing that our simulation-trained policies generate stable, reliable, and directly transferable motion trajectories for real hardware operation. While constraints inherent to operating precision surgical robotics hardware—including stringent safety protocols and limited availability—precluded large-scale quantitative testing in this initial deployment phase, the observed successful completion of all task objectives provides critical proof-of-concept validation. This demonstration confirms that the behaviors learned in simulation can be effectively transferred to physical systems, thereby verifying the practical viability of our entire pipeline for real-world surgical applications.

E. Discussion and Limitations

While our synthetic action-level and trajectory-level perturbations are designed to emulate real-world sensor noise and common cognitive errors (e.g., mis-grasping or reaching the wrong goal), it remains an open question whether these non-optimal trajectories fully capture the complexity and biomechanical variations of realistic surgeon behaviors. Future work will focus on collecting and incorporating ac-

tual surgeon-generated suboptimal demonstrations to further validate the realism of our perturbations.

V. CONCLUSIONS

We propose a diffusion-based policy learning framework, called DSP, that enables training a diffusion model using a mixture of clean demonstrations and perturbed trajectories. We showcase the superior performance of our method in the absence of perturbations and its robustness when handling perturbed data. We hope our approach serves as a step towards applying diffusion policy on surgical tasks, sparking further interest in this direction. Moreover, we hope our method paves the way for scaling up data in the field of surgical robotics.

APPENDIX

A. Action-level Perturbation Settings

We considered four types of perturbation noise added to expert actions a^*:

Gaussian noise: A random perturbation $\epsilon \sim \mathcal{N}(\eta, \sigma)$ with mean $\eta = 0.05$ and standard deviation $\sigma = 0.2$, applied independently to each action dimension with probability 0.5.

Uniform noise: A perturbation $\epsilon \sim \mathcal{U}(-0.2, 0.2)$, applied to each dimension with probability 0.5.

Poisson noise: A perturbation $\epsilon \sim \text{Poisson}(\lambda)$ with rate parameter $\lambda = 0.2$. The perturbation is applied with probability 0.5.

B. Trajectory-level Perturbation Settings

This appendix provides a comprehensive classification and description of imperfect demonstration types implemented for each surgical task in simulations. These non-optimal trajectories represent common execution failures that may occur during robotic surgical procedures, providing valuable data for training robust imitation learning algorithms. If not stated, these imperfect demonstrations are implemented by adding way-points for each task during the data generating procedure. In the table, the type numbering of trajectory-level noise corresponds to the order in the following list.

1) PSM Tasks:

a) NeedlePick: **double grasp**: Requires two attempts to successfully secure the needle. **didnt grasp**: Fails to grip the needle despite proper positioning. **wrong goal**: Succeeds in grasping but navigates to an incorrect target.

b) NeedleReach: **didn't reach**: Fails to achieve the target coordinates. **bad reach**: first approaches the vicinity, then fine-tuning to the final destination.

c) PegTransfer: **bad drag**: Suboptimal transportation path with excessive movement via an added way-point. **double grasp**: Requires a secondary attempt to pick up the peg. **miss drop**: Fails to deploy the peg at the target.

d) GauzeRetrieve: **double grasp**: Requires two attempts to secure the gauze. **didn't grasp**: Fails to acquire the gauze but continues the remaining movement. **wrong goal**: Retrieves the gauze but transports it to an incorrect destination.

2) Bi-PSM Tasks (Bimanual Manipulation):

a) BiPegTransfer: **bad transfer**: Suboptimal coordination between manipulators during object transfer, characterized by awkward kinematic configurations.

double grasp: Requirement for two grasping attempts by the grasping manipulator during the transfer process.

miss drop: Failure to accurately deploy the peg at the designated target zone.

b) NeedleRegrasp: **bad transfer**: Poorly coordinated needle exchange between manipulators, demonstrating suboptimal bi-manual coordination.

didn't grasp: Failure to successfully transfer needle ownership between manipulators, while the PSMs perform the remaining movements.

wrong goal: Successful needle regrasping followed by navigation to an incorrect target location.

C. Implementation Details of Adaptive Thresholding Methods

To distinguish clean from noisy samples, we compute thresholds on log-transformed errors (to balance magnitudes) and map them back to the original error scale:

1) Otsu's Method: : A histogram with 40 bins in log-space was built. The threshold was chosen as the value that maximized between-class variance, then mapped back to the original error scale.

2) Gaussian Mixture Model (GMM): : A two-component GMM from scikit-learn was fit to the log-errors, with means initialized at the 10th and 90th percentiles. Training ran up to 500 iterations with covariance regularization (1e-6). The threshold was the intersection where both components had equal posterior probability, then converted back to the original scale.

3) K-means Clustering: : Midpoint between two cluster centers computed via scikit-learn's KMeans.

ACKNOWLEDGMENT

This work has been supported in part by the program of National Natural Science Foundation of China (No.62503322), Shanghai Magnolia Funding Pujiang Program (No. 23PJ1404400), and the AI for Science Seed Program of Shanghai Jiao Tong University (project number 2025AI4SQY06).

The work from L. Song and Q. Dou has been supported in part by a grant from the NSFC/RGC Joint Research Scheme sponsored by the Research Grants Council of the Hong Kong Special Administrative Region, China and the National Natural Science Foundation of China (Project No. N CUHK410/23), and in part by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. 14208424).

REFERENCES

[1] K. L. Schwaner, D. Dall'Alba, P. T. Jensen, P. Fiorini, and T. R. Savarimuthu, "Autonomous needle manipulation for robotic surgical suturing based on skills learned from demonstration," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, 2021, pp. 235–241.

[2] F. Zhong, Y. Wang, Z. Wang, and Y.-H. Liu, "Dual-arm robotic needle insertion with active tissue deformation for autonomous suturing," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2669–2676, 2019.

[3] H. Saeidi, J. D. Opfermann, M. Kam, S. Wei, S. Leonard, M. H. Hsieh, J. U. Kang, and A. Krieger, "Autonomous robotic laparoscopic surgery for intestinal anastomosis," *Science Robotics*, vol. 7, no. 62, p. eabj2908, 2022.

[4] A. Pore, D. Corsi, E. Marchesini, D. Dall'Alba, A. Casals, A. Farinelli, and P. Fiorini, "Safe reinforcement learning using formal verification for tissue retraction in autonomous robotic-assisted surgery," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4025–4031.

[5] J. J. Ji, S. Krishnan, V. Patel, D. Fer, and K. Goldberg, "Learning 2d surgical camera motion from demonstrations," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. IEEE Press, 2018, p. 35–42.

[6] X. Ma, C. Song, P. W. Chiu, and Z. Li, "Autonomous flexible endoscope for minimally invasive surgery with enhanced safety," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2607–2613, 2019.

[7] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky, "π_0: A vision-language-action flow model for general robot control," 2024.

[8] Embodiment Collaboration, A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, A. Tung, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Gupta, A. Wang, ..., and Z. Lin, "Open x-embodiment: Robotic learning datasets and rt-x models," 2024.

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

[10] J. W. Kim, P. Zhang, P. Gehlbach, I. Iordachita, and M. Kobilarov, "Towards autonomous eye surgery by combining deep imitation learning with optimal control," in *Proceedings of the 2020 Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. Tomlin, Eds., vol. 155. PMLR, 16–18 Nov 2021, pp. 2347–2358.

[11] B. Li, R. Wei, J. Xu, B. Lu, C.-H. Yee, C.-F. Ng, P.-A. Heng, Q. Dou, and Y.-H. Liu, "3d perception based imitation learning under limited demonstration for laparoscope control in robotic surgery," 2022.

[12] P. M. Scheikl, N. Schreiber, C. Haas, N. Freymuth, G. Neumann, R. Lioutikov, and F. Mathis-Ullrich, "Movement primitive diffusion: Learning gentle robotic manipulation of deformable objects," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5338–5345, 2024.

[13] K. Kawaharazuka, K. Okada, and M. Inaba, "Robotic constrained imitation learning for the peg transfer task in fundamentals of laparoscopic surgery," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2024, p. 606–612.

[14] J. W. Kim, T. Z. Zhao, S. Schmidgall, A. Deguet, M. Kobilarov, C. Finn, and A. Krieger, "Surgical robot transformer (srt): Imitation learning for surgical tasks," 2024.

[15] J. Xu, B. Li, B. Lu, Y.-H. Liu, Q. Dou, and P.-A. Heng, "Surrol: An open-source reinforcement learning centered and dvrk compatible platform for surgical robot learning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 1821–1828.

[16] P. M. Scheikl, B. Gyenes, R. Younis, C. Haas, G. Neumann, M. Wagner, and F. Mathis-Ullrich, "Lapgym – an open source framework for reinforcement learning in robot-assisted laparoscopic surgery," 2023.

[17] J. Wu, H. Zhou, P. Kazanzides, A. Munawar, and A. Liu, "Surgicai: A hierarchical platform for fine-grained surgical policy learning and benchmarking," 2025.

[18] T. Huang, K. Chen, B. Li, Y. Liu, and Q. Dou, "Demonstration-guided reinforcement learning with efficient exploration for task automation of surgical robot," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4640–4647, 2023.

[19] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020.

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

[21] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10 674–10 685, 2021.

[22] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," *ArXiv*, vol. abs/2403.03954, 2024.

[23] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu, "Rdt-1b: a diffusion foundation model for bimanual manipulation," 2024.

[24] Y. Long, A. Lin, D. H. C. Kwok, L. Zhang, Z. Yang, K. Shi, L. Song, J. Fu, H. Lin, W. Wei, K. Chen, X. Chu, Y. Hu, H. C. Yip, P. W. Y. Chiu, P. Kazanzides, R. H. Taylor, Y. Liu, Z. Chen, Z. Wang, null, and Q. Dou, "Surgical embodied intelligence for generalized task autonomy in laparoscopic robot-assisted surgery," *Science Robotics*, vol. 10, no. 104, p. eadt3093, 2025. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.adt3093

[25] G. Li, Z. Jin, M. Volpp, F. Otto, R. Lioutikov, and G. Neumann, "Prodmp: A unified perspective on dynamic and probabilistic movement primitives," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2325–2332, 2023.

[26] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International Conference on Machine Learning (ICML)*, 2018.

[27] G. Daras, K. Shah, Y. Dagan, A. Gollakota, A. Dimakis, and A. Klivans, "Ambient diffusion: Learning clean distributions from corrupted data," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[28] B. Na, Y. Kim, H. Bae, J. H. Lee, S. J. Kwon, W. Kang, and I.-C. Moon, "Label-noise robust diffusion models," in *ICLR*, 2024.

[29] Z. Dong, Y. Yuan, J. HAO, F. Ni, Y. Ma, P. Li, and Y. ZHENG, "Cleandiffuser: An easy-to-use modularized library for diffusion models in decision making," in *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[30] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2017.

[31] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning." in *International Conference on Learning Representations (ICLR)*, 2016.

[32] M. Bain and C. Sammut, "A framework for behavioural cloning," in *Machine Intelligence*, 1999.

[33] S. Reddy, A. D. Dragan, and S. Levine, "Sqil: Imitation learning via reinforcement learning with sparse rewards," in *International Conference on Learning Representations (ICLR)*, 2019.

[34] J. Pari, N. M. M. Shafiullah, S. P. Arunachalam, and L. Pinto, "The surprising effectiveness of representation learning for visual imitation," in *Robotics: Science and Systems (RSS)*, 2022.

[35] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[36] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, "Amp: Adversarial motion priors for stylized physics-based character control," *ACM Transactions on Graphics (ToG)*, 2021.

[37] V. G. Goecks, G. M. Gremillion, V. J. Lawhern, J. Valasek, and N. R. Waytowich, "Integrating behavior cloning and reinforcement learning for improved performance in dense and sparse reward environments," in *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2020, pp. 465–473.

[38] A. Nair, A. Gupta, M. Dalal, and S. Levine, "Awac: Accelerating online reinforcement learning with offline datasets," *arXiv preprint arXiv:2006.09359*, 2020.

[39] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[40] V. Melnykov and R. Maitra, "Finite mixture models and model-based clustering," 2010.

[41] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the royal statistical society. series c (applied statistics)*, vol. 28, no. 1, pp. 100–108, 1979.