Sim-to-Real Diffusion Policies for Robotic Alignment in Manufacturing

Yikai Zheng[1], Zhaohang Zheng[1],Jason Gu[2] and Wei Liu[1]3

***Abstract*— Robust robotic alignment is a key capability in manufacturing and offshore maintenance, where precision components must be positioned under uncertain conditions. However, existing solutions are often labor-intensive or rely on rigid manipulation assumptions that limit their applicability.**

In this paper, we propose a push-based manipulation framework integrating imitation learning with simulation-to-real (Sim2Real) transfer. A diffusion-based policy is trained from expert demonstrations to generate temporally coherent multi-step action sequences for contact-rich and stochastic interactions.

To bridge the reality gap, we introduce a semantic-consistent visual preprocessing pipeline that aligns real-world observations with simulation via segmentation and recoloring, improving robustness under lighting variations and occlusions.

Experiments in simulation and real-world settings show that the proposed method achieves high success rates and strong robustness across diverse geometries and perturbations, demonstrating the effectiveness of diffusion-based visuomotor policies for reliable deployment in marine and industrial alignment tasks.

***Index Terms*— Robotic alignment, imitation learning, Sim2Real transfer, push-based manipulation,**

I. Introduction

Robotic alignment is a critical operation in manufacturing and offshore maintenance, where precision components must be positioned accurately to ensure reliable system performance. In such environments, alignment tasks are often performed manually, leading to low efficiency and inconsistent quality. Automating these processes remains challenging due to uncertain contact dynamics, perception noise, and environmental variability.

Robotic automation provides a promising alternative, yet existing approaches face significant limitations. Grasping-based methods struggle with smooth or delicate surfaces due to slipping and potential damage, while suction-based solutions require strict geometric constraints [1], [2]. These limitations restrict their applicability across diverse insert types.

Push-based manipulation offers a more flexible alternative by enabling incremental adjustments without requiring stable grasps [3]. This approach better reflects human alignment strategies and is particularly effective for handling objects with irregular shapes or uncertain physical properties. However, existing approaches fail to simultaneously achieve robust perception and efficient policy transfer in contact-rich manipulation scenarios. Traditional push-based methods often rely heavily on visual feedback, making them vulnerable to occlusions and perception failures.

To address these challenges, we propose a framework that integrates imitation learning with simulation-to-reality (Sim2Real) transfer [4], [5] for robust push-based alignment. By learning directly from expert demonstrations, the system captures strategies for handling occlusions and dynamic interactions, reducing reliance on precise visual perception.

As illustrated in Fig. 1, our method builds upon diffusion-based policy learning [6] to generate multi-step action sequences, while a semantic segmentation module enforces visual consistency between simulated and real-world observations, facilitating effective sim-to-real transfer.

The main contributions of this work are:

- A push-based manipulation framework that combines imitation learning with Sim2Real transfer for CNC insert alignment.
- A diffusion-based policy that generates multi-step action sequences from expert demonstrations for robust and adaptive manipulation.
- A semantic segmentation-based domain adaptation strategy that improves visual consistency across simulation and real-world environments.
- Extensive validation in both simulated and real-world settings, demonstrating improved alignment accuracy and robustness across diverse insert geometries.

Although our experiments focus on laboratory CNC insert alignment, the challenges in this task closely resemble those faced by offshore and marine robotic operations. In such environments, robots often need to align and install components in constrained spaces while dealing with uncertain contact dynamics and variable visual conditions. The incremental push-based strategy studied here mirrors the approach required for delicate adjustments in marine maintenance tasks, such as valve alignment or hose installation on offshore platforms.

II. Related Work

A. Imitation Learning for Robot Manipulation

Recent advances in diffusion models [7], [8] have significantly improved sequential decision-making [6], [9] by enabling multi-modal trajectory generation. Prior works have

This research was supported by the 2023 Key Project of Guangdong Provincial Department of Education for General Universities (Project No. 2023ZDZX3024), the 2023 Guangdong Basic and Applied Basic Research Fund Regional Joint Fund-Key Project (Project No. 2023B1515120017), and the National Foreign Experts Program (Project No. S20240047).

[1]Yikai Zheng, Zhaohang Zheng, and Wei Liu are with the Department of Mechanical and Energy Engineering, Southern University of Science and Technology, Shenzhen 518055, China.

[2]Jason Gu is with the Department of Electrical and Computer Engineering, Dalhousie University, Halifax, NS, Canada.

[3]Wei Liu is with Advanced Institute for Ocean Research, Southern University of Science and Technology, Shenzhen, 518055, China

Corresponding author: Wei Liu (email: liuw2@sustech.edu.cn).

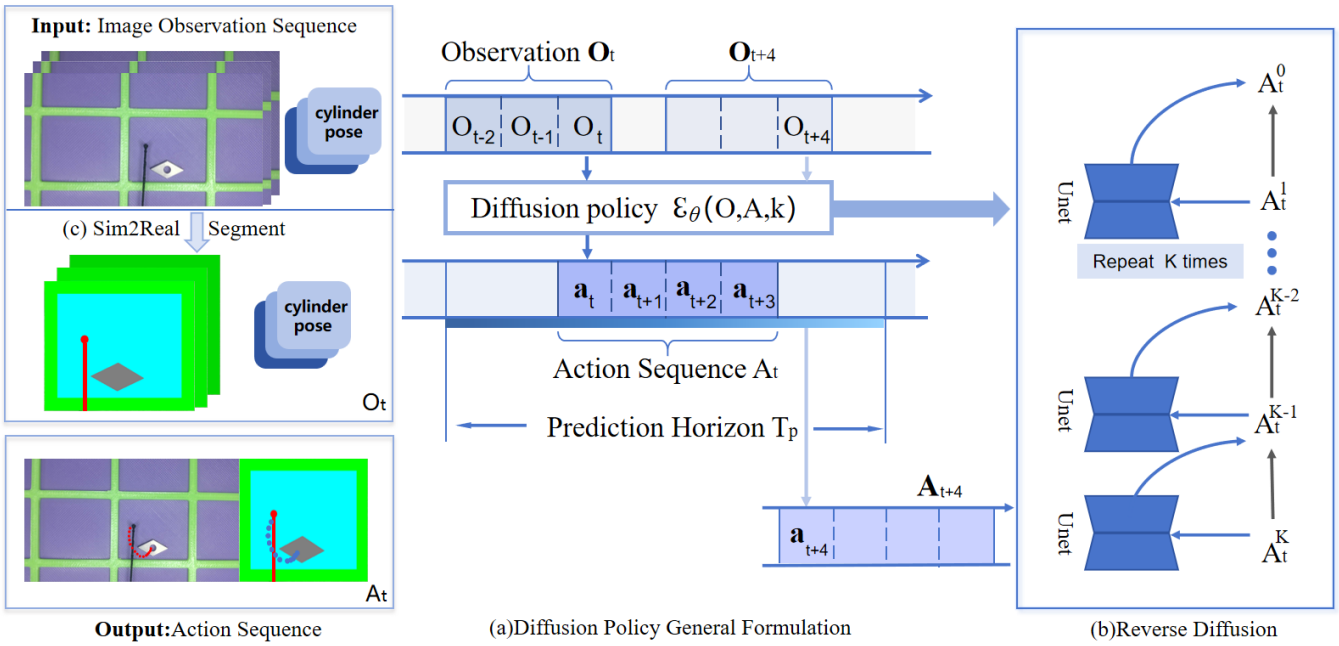

Fig. 1. **Object Pushing Framework:** (a) Policy formulation. At time step t, the policy takes the latest T_o observations $\mathcal{O}_t$ and outputs T_a actions $\mathcal{A}_t$. (b) Reverse diffusion. Starting from Gaussian noise $\mathcal{A}_t^K$, the noise prediction network ϵ_θ iteratively denoises the sequence to obtain $\mathcal{A}_t^0$ [6]. (c) Sim2Real module. A U-Net-based semantic segmentation network extracts key elements (manipulator, insert, and workspace) to ensure visual consistency across domains.

demonstrated strong performance in robotic manipulation tasks, particularly in contact-rich and stochastic environments [6]. The iterative denoising process allows diffusion policies to capture diverse feasible solutions.

However, diffusion-based policies typically require large-scale datasets and face challenges when deployed directly in real-world environments due to distribution mismatch.

B. Simulation-to-Real Transfer

Simulation-to-real (Sim2Real) transfer has been widely adopted to address the reality gap in robotic learning. Domain randomization [10] improves robustness by introducing variability in simulation, while domain adaptation methods [11], [12] align feature distributions between simulated and real environments.

Although effective, these approaches may still struggle in dynamic manipulation tasks where accurate modeling of physical interactions is difficult.

C. Sim2Real for Imitation Learning

Combining imitation learning with Sim2Real transfer has shown promise in improving data efficiency and reducing reliance on real-world data [13], [4]. In addition, structured control priors and kinematic randomization have been explored to improve transfer stability [14].

Nevertheless, the integration of diffusion-based policies with Sim2Real remains limited. In particular, efficiently leveraging simulated data while maintaining robustness to real-world uncertainties is still an open challenge, especially for contact-rich tasks such as pushing [3], [15]. Compared to prior work, our approach explicitly enforces semantic visual consistency between domains, enabling more reliable Sim2Real transfer for diffusion-based policies in contact-rich manipulation tasks.

III. Method

A. Diffusion Policy for Imitation Learning

We formulate the pushing policy as a conditional generative model that predicts a sequence of future actions given the current observation. Following [6], the policy models the conditional distribution of a multi-step action sequence $\mathbf{a}_{1:H}$ conditioned on $\mathbf{o}_t$. An overview of the framework is shown in Fig. 1.

Action sequence prediction: At each timestep t, the policy takes the most recent T_o observations as input and predicts T_p future actions. Instead of executing only the first action, we adopt a receding-horizon control strategy by executing the first T_a actions before replanning. This design improves temporal consistency while maintaining adaptability to new observations. Such a formulation allows the policy to implicitly reason over short-term future interactions, which is critical for contact-rich pushing tasks requiring sequential adjustments.

Diffusion formulation: To model complex and multi-modal action distributions, we adopt a denoising diffusion process. The forward process gradually perturbs expert action sequences with Gaussian noise:

$$q(\mathbf{a}_t^k \mid \mathbf{a}_t^{k-1}) = \mathcal{N}\left(\sqrt{1-\beta_k}\,\mathbf{a}_t^{k-1},\, \beta_k \mathbf{I}\right), \tag{1}$$

where $\mathbf{a}_t^0$ is the original expert action sequence and $\mathbf{a}_t^k$ is the noisy version after k steps.

The reverse process reconstructs clean actions by iteratively removing noise:

$$\mathbf{a}_t^{k-1} = \alpha_k \Big(\mathbf{a}_t^k - \gamma_k \, \epsilon_\theta(\mathbf{o}_t, \mathbf{a}_t^k, k) \Big) + \mathcal{N}(0, \sigma_k^2 I), \quad (2)$$

where ϵ_θ is a learnable noise prediction network, and $\alpha_k, \gamma_k, \sigma_k$ are derived from the noise schedule.

The training objective minimizes the discrepancy between true noise and predicted noise:

$$\mathcal{L} = \mathrm{MSE}\big(\epsilon^k, \, \epsilon_\theta(\mathbf{o}_t, \mathbf{a}_t^0 + \epsilon^k, k)\big) \, . \quad (3)$$

Architecture details: As illustrated in Fig. 2, the policy adopts a temporal U-Net architecture that takes noisy action sequences and observation-conditioned features as input. The encoder consists of stacked 1D convolutional layers with residual connections to capture long-range temporal dependencies, while FiLM layers inject observation conditioning at each level. The decoder mirrors the encoder via upsampling and skip connections to reconstruct temporally consistent action sequences.

Discussion: Compared to single-step policies, the diffusion formulation captures multi-modal action distributions, which is critical for contact-rich manipulation tasks with uncertain dynamics. The iterative denoising process promotes smooth and temporally consistent action generation, improving interaction stability.

Furthermore, multi-step prediction enables implicit short-horizon planning, allowing the policy to recover from intermediate errors and adapt to dynamic changes. This is particularly important in pushing tasks, where early deviations can significantly affect final alignment performance.

B. Semantic-Consistent Sim-to-Real Transfer

To reduce the visual domain gap between simulation and real-world environments, we design a semantic-consistent preprocessing pipeline that transforms real observations into a simulation-aligned representation.

Pipeline overview: Given an RGB image, we first localize the workspace region using template matching [16]. A U-Net-based model [17] is then applied to segment key components, including the CNC insert, manipulator, and workspace.

Semantic alignment: The segmented regions are mapped to a predefined color space consistent with the simulation environment, resulting in a standardized representation used as policy input.

Rationale: This approach reduces domain discrepancy at the input level, allowing the diffusion policy to operate on observations that closely match the training distribution, thereby improving robustness to illumination variation and partial occlusion.

C. Data Collection and Training

Data collection: Demonstration data are collected in simulation via manual teleoperation. A human operator controls the end-effector using mouse input, enabling intuitive generation of expert trajectories.

The dataset includes diverse initial configurations, with variations in object position, orientation, and contact conditions. This diversity is essential for learning robust manipulation strategies.

Action generation: The action space is defined as planar displacements $\mathbf{a}_t = (\Delta x_t, \Delta y_t)$. Mouse movement between consecutive frames is recorded and scaled to generate smooth and continuous actions in the simulation environment.

Data recording: At each timestep, we record:

- RGB observations from the virtual camera,
- Agent states including position and velocity,
- Executed action $\mathbf{a}_t$.

These data are stored sequentially to form complete trajectories.

Training: The collected dataset of $(\mathbf{o}_{1:T}, \mathbf{a}_{1:T})$ pairs is used to train the policy via the diffusion policy framework [6]. As described in model architecture, the policy models the conditional distribution $p(\mathbf{a}_{1:H} \mid \mathbf{o}_t)$ of multi-step future actions given the current observation. In training, the forward diffusion process progressively adds Gaussian noise to the expert action sequences, and the denoising network ϵ_θ is optimized via backpropagation to reconstruct the original actions by predicting the noise at each step. The loss function follows the denoising score matching objective introduced in DDPM [7]. The model learns to reconstruct clean action sequences conditioned on observations.

Inference and execution: At test time, the policy samples action sequences by iteratively denoising from Gaussian noise. Instead of executing a single action, we adopt a block execution strategy:

$$\{\hat{\mathbf{a}}_t, \ldots, \hat{\mathbf{a}}_{t+\kappa-1}\}.$$

After executing κ steps, the system re-observes the environment and replans, balancing stability and adaptability.

IV. Experiments and Results

A. Simulation Setup

We construct a physics-based simulation environment in PyBullet to train and evaluate the policy. The environment includes multiple CNC insert geometries with randomized initial positions and orientations to improve generalization.

The agent interacts with the environment through planar actions $\mathbf{a}_t \in \mathbb{R}^2$, and observations consist of RGB images and agent states.

Simulation details: The simulation is implemented in PyBullet with a fixed timestep of $\Delta t = 1/240$ s. The cylindrical pushing tool is controlled via a velocity-based controller with saturation to ensure stable contact interactions.

To improve robustness and generalization, we apply domain randomization during training, which has been widely shown to improve sim-to-real transfer performance [10], [5]. Object initial positions are uniformly sampled within a bounded workspace, orientations are randomly generated, and friction coefficients are randomized within a predefined range.

A fixed overhead camera generates RGB observations, resized to 256×256 pixels as policy input.

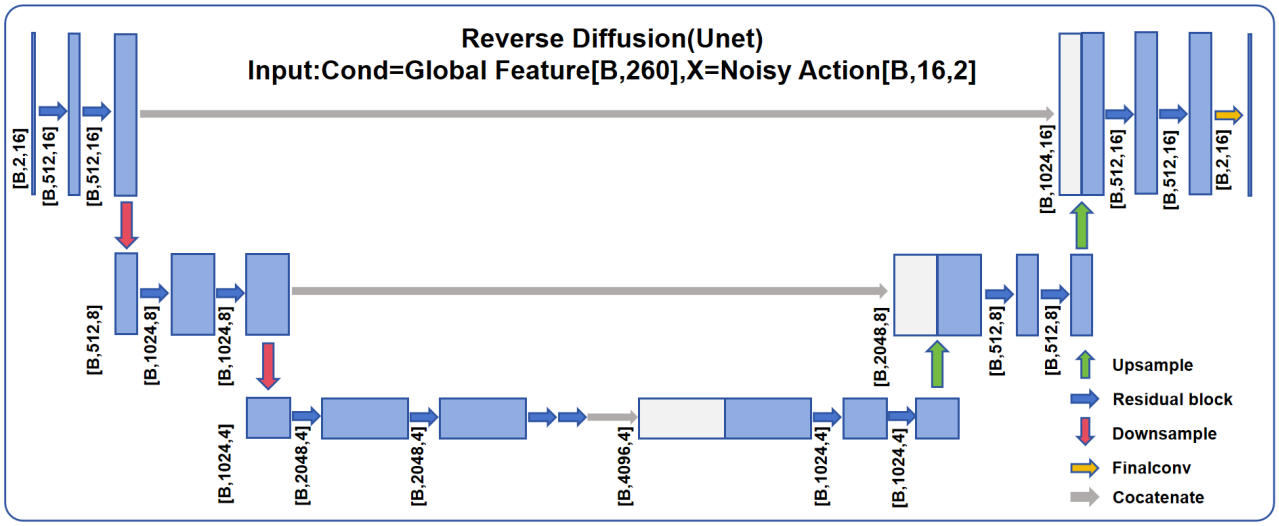

Fig. 2. **U-Net architecture of the diffusion policy.**The network takes noisy action sequences and observation-conditioned features as input and iteratively denoises them to produce executable action sequences.

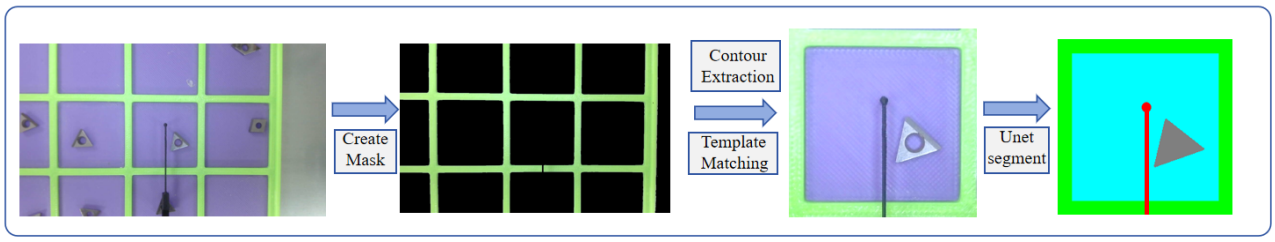

Fig. 3. **Sim-to-Real visual preprocessing pipeline.**

B. Task and Evaluation Metrics

The task requires pushing a CNC insert to a target pose using a cylindrical rod. The agent must jointly regulate position and orientation through planar pushing actions, which is closely related to classical planar manipulation and pushing dynamics [3], [15].

Performance is evaluated using:

- **Success rate**: percentage of episodes achieving successful alignment.
- **Final reward**: cumulative reward per episode.
- **Steps**: number of steps required to complete the task.

An episode is considered successful if both position and orientation errors fall below predefined thresholds within a maximum horizon.

Reward definition: The reward function is defined as:

$$r_t = -\left(w_p \epsilon_p + w_\theta \epsilon_\theta\right) + r_{\text{time}} + r_{\text{succ}}, \tag{4}$$

where ϵ_p and ϵ_θ denote position and orientation errors. The time penalty is:

$$r_{\text{time}} = -0.1. \tag{5}$$

The success reward is:

$$r_{\text{succ}} = \begin{cases} R_s, & \epsilon_p < \delta_p \ \wedge \ \epsilon_\theta < \delta_\theta \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

This formulation is consistent with sparse success signals commonly used in robotic manipulation learning [6], [1].

with $w_p = 0.7$, $w_\theta = 0.3$, $\delta_p = 8$ px, and 6° (≈ 0.105 rad).

Success criterion:

$$\epsilon_p < \delta_p, \quad \epsilon_\theta < \delta_\theta. \tag{7}$$

C. Policy Learning in Simulation

Training details: The policy is trained using the Adam optimizer [18] with learning rate 1×10^{-4} and cosine decay. Batch size is 32, and training runs for 200k iterations.

We adopt a diffusion-based policy, which models action sequences as a generative process and has shown strong performance in visuomotor learning tasks [6], [7], [8], [9].

We use a prediction horizon of 16 steps and execute 8 actions per rollout before replanning.

As shown in Fig. 4, the success rate increases significantly after 100k steps, reaching 0.96.

D. Real-World Experiment Setup

We deploy the proposed method on a three-axis linear manipulation platform equipped with a monocular vision system, as shown in Fig. 5. The system is designed for high-precision planar pushing tasks and enables stable sim-to-real evaluation.

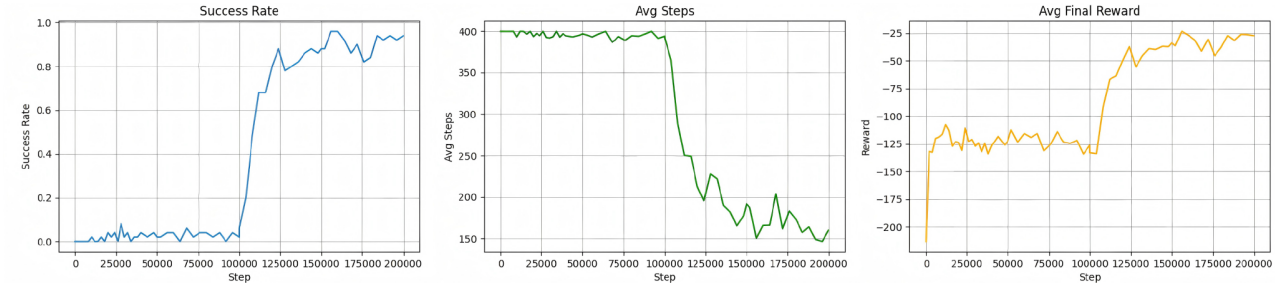

Fig. 4. Training performance over time.

Three-axis linear platform: The experimental platform provides a $300 \times 300 \times 300$ mm workspace with an orthogonal mechanical structure. Motion is actuated by stepper motors coupled with ball-screw transmissions, achieving a positioning accuracy of approximately 0.1 mm in the XY plane.

To ensure smooth motion, trapezoidal velocity profiles are used, a standard technique in robotic actuation systems. Closed-loop feedback is achieved using incremental encoders.

The mechanical structure is built from aluminum profiles, providing high rigidity while maintaining low inertia.

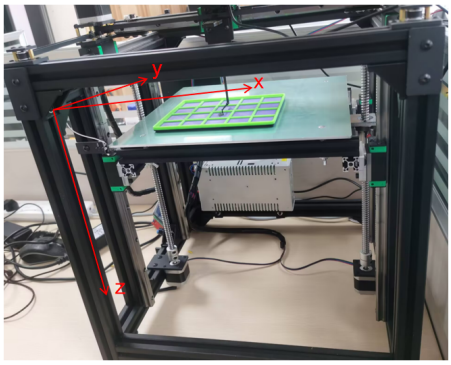

Fig. 5. Real-world experimental platform.

Vision system: A fixed monocular RGB camera provides real-time visual feedback. Images are resized to 256×256 as policy input.

For perception, we adopt a U-Net-based segmentation model [17], which is widely used in dense prediction tasks. Edge and structure priors are further extracted using classical vision operators such as Canny edge detection [19] and Hough transform [20].

E. Real Robot Experiment

As shown in Fig. 7, the proposed policy achieves consistently high success rates across different CNC insert types, with all cases exceeding 80%. Each result is evaluated over 30 independent trials per insert type.

During real-world deployment, we observed that failures were mainly caused by:

- Occluded or mis-segmented inserts.
- Initial object placement near the workspace edges.

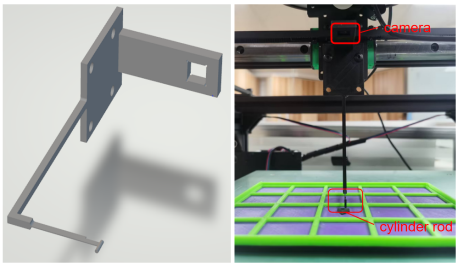

Fig. 6. Camera module.

These observations emphasize the importance of maintaining visual consistency and generating robust multi-step actions for effective sim-to-real transfer. Our method employs semantic preprocessing and recoloring to reduce the visual domain gap between simulation and reality. While we did not perform a full quantitative ablation study in this work, our limited trials and prior literature suggest that removing semantic preprocessing or recoloring can significantly degrade performance [10], [5]. Similarly, baseline policies without diffusion modeling struggle to plan robust multi-step action sequences under uncertain contact dynamics [6].

The reward and step statistics exhibit a moderate long-tail distribution. Most trials achieve medium-to-high performance, while a small number of edge cases result in lower rewards and increased execution steps. Overall, the policy demonstrates strong stability and repeatability in real-world, contact-rich manipulation tasks, with occasional performance degradation under challenging initial conditions.

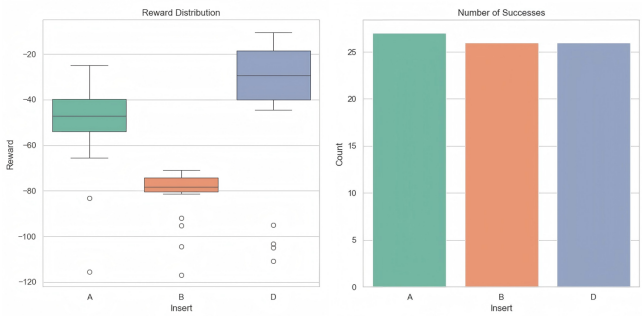

Fig. 7. Push task results of A, B, and D CNC inserts. Success rates are 90.0%, 86.7%, and 83.3%. Each result is evaluated over 30 independent trials per insert type.

Robustness Test: To further evaluate robustness under external disturbances, we introduce perturbations at different stages of task execution. As shown in Fig. 8, two representative scenarios are considered: late-stage disturbance near task completion and mid-trajectory disturbance during execution.

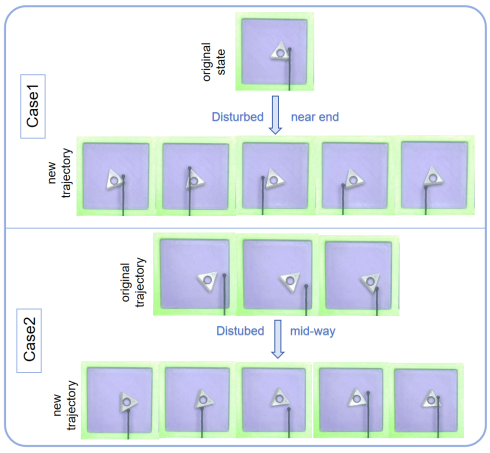

Fig. 8. Robustness evaluation under external disturbances. Case 1: late-stage disturbance. Case 2: mid-trajectory disturbance.

In the first case, the policy is able to re-localize the object after a late-stage disturbance and complete the task, although with increased execution steps in some trials. In the second case, despite mid-trajectory perturbations, the policy demonstrates strong recovery capability by adjusting its actions and continuing toward the goal. Overall, the results show that the proposed method maintains reliable performance under dynamic disturbances, while incurring additional recovery cost in more challenging scenarios.

V. Conclusion

We proposed a diffusion-policy-based imitation learning framework with a semantic-consistent Sim2Real pipeline for robotic object alignment in marine and industrial manufacturing scenarios.

Experiments in simulation and on a real-world platform show high success rates and robust performance under varying initial conditions and disturbances. Failures mainly stem from perception errors, challenging initial placements, or tool flexibility.

Future work will focus on:

1) Systematic ablation studies to quantify the contributions of semantic preprocessing, recoloring, and diffusion-based policy.
2) Improving visual robustness against occlusion and illumination changes.
3) Extending to more complex, contact-rich manipulation tasks.

The results demonstrate the potential of combining diffusion-based policies with semantic-consistent Sim2Real transfer for reliable real-world robotic alignment.

References

[1] A. Lobbezoo, Y. Qian, and H.-J. Kwon, "Reinforcement learning for pick and place operations in robotics: A survey," *Robotics*, vol. 10, no. 3, 2021.

[2] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Dex-net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5620–5627.

[3] K.-T. Yu, M. Bauza, N. Fazeli, and A. Rodriguez, "More than a million ways to be pushed. a high-fidelity experimental dataset of planar pushing," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 30–37.

[4] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis, "Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 619–12 629.

[5] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2018, p. 3803–3810.

[6] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[7] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851.

[8] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2021.

[9] M. Janner, Y. Du, J. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *Proceedings of the 39th International Conference on Machine Learning*, vol. 162, 17–23 Jul 2022, pp. 9902–9915.

[10] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 23–30.

[11] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 59, pp. 1–35, 2016.

[12] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "CyCADA: Cycle-consistent adversarial domain adaptation," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, 10–15 Jul 2018, pp. 1989–1998.

[13] A. A. Rusu, M. Večerík, T. Rothörl, N. Heess, R. Pascanu, and R. Hadsell, "Sim-to-real robot learning from pixels with progressive nets," in *Proceedings of the 1st Annual Conference on Robot Learning*, vol. 78, 13–15 Nov 2017, pp. 262–270.

[14] I. Exarchos, Y. Jiang, W. Yu, and C. Karen Liu, "Policy transfer via kinematic domain randomization and adaptation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 45–51.

[15] S. Goyal, A. Ruina, and J. Papadopoulos, "Planar sliding with dry friction part 2. dynamics of motion," *Wear*, vol. 143, no. 2, pp. 331–352, 1991.

[16] J.-C. Yoo and T. H. Han, "Fast normalized cross-correlation," *Circuits, Systems and Signal Processing*, vol. 28, pp. 819–843, 2009.

[17] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Cham, 2015, pp. 234–241.

[18] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[19] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[20] N. Kiryati, Y. Eldar, and A. Bruckstein, "A probabilistic hough transform," *Pattern Recognition*, vol. 24, no. 4, pp. 303–316, 1991.