Desktop Cleaning by Robotic Arm via 3D Diffusion Policy

Yitao Zeng
School of System Design and Intelligent Manufacturing
Southern University of Science and Technology
Shenzhen, China
12312903@mail.sustech.edu.cn

Yufeng Wen
School of System Design and Intelligent Manufacturing
Southern University of Science and Technology
Shenzhen, China
12211423@mail.sustech.edu.cn

Chuyao Fu
Department of Electronic and Electrical Engineering
Southern University of Science and Technology
Shenzhen, China
12310306@mail.sustech.edu.cn

***Abstract*—Developing robotic systems capable of performing diverse manipulation tasks in various daily scenarios has long been a research objective. Traditional approaches that integrate visual localization, trajectory planning, and control algorithms enable precise manipulation of predefined objects, yet face significant limitations. These methods struggle to adapt to real-world task diversity, demonstrate limited stability against sudden disturbances, and typically require specific visual markers (e.g., ArUco codes) for reliable operation.**

In contrast, imitation learning algorithms offer superior environmental adaptability and stability, with the added advantage of easy transferability across multiple tasks through appropriate training data. This paper implements the 3D Diffusion Policy, an imitation learning framework based on diffusion models, to enable autonomous desktop clearing by robotic arms. Our research contributions include: (1)Establishing a complete technological pipeline encompassing data collection, policy training, and deployment execution, enabling rapid learning of new tasks through framework reuse without code modifications; (2)Achieving scene-adaptive performance using only depth camera information, eliminating dependence on external localization markers; and (3)Incorporating model predictive control principles to enhance policy robustness.

Experimental results confirm the framework's effectiveness in robotic manipulation tasks, while providing practical insights for engineering applications of diffusion policy frameworks in unstructured environments. The project code is publicly available at: https://github.com/Undefinedefity/ARX_dp3_training ,https://github.com/ChuyaoZellFu/ARX-Remote-Controland https://github.com/24zen0/depth-to-pointcloud-for-dp3

***Index Terms*—Robotic arm manipulation, Diffusion policy, Imitation learning, Desktop Cleaning**

I. Introduction

The core technical challenge in the field of embodied intelligence is establishing an effective mapping from spatial perception to action control, which has attracted extensive academic research for a long time. With the widespread application of robotic systems in various fields such as industrial automation, smart homes, and medical assistance, the development of robotic manipulation technologies adaptable to complex daily scenarios has become particularly urgent. In industrial production, robots are required to precisely perform tasks such as component assembly and handling; in smart home scenarios, robots need to achieve functions like item organization and environmental cleaning; in the medical field, tasks such as robot - assisted surgery and rehabilitation nursing demand extremely high precision and stability. However, although traditional rule - based approaches can maintain stability in structured environments, they rely on high - precision localization tokens (e.g., Aruco codes) or complex manual rule design. These localization tokens are prone to failure under conditions such as lighting changes and occlusions, and manual rule design is difficult to cover all possible complex situations. When facing sudden perturbations such as abrupt changes in object positions, sensor occlusions, and environmental layout changes, traditional methods lack robustness, making it difficult to directly apply them to unstructured daily life scenarios.

In recent years, Diffusion Policy has emerged as a cutting - edge research direction in the field of robot manipulation due to its strong generalization ability, real - time reasoning efficiency, and natural adaptability to environmental perturbations. By modeling the probability distribution of action sequences, this method can learn complex task patterns from limited operational data, overcoming the drawbacks of traditional methods that require customized code or long trial - and - error training. Take the desktop clearing task as an example. Traditional methods may require writing a large amount of specific code for different desktop item arrangements, shapes, and materials, while the Diffusion Policy can master the general pattern of the clearing task only through learning a small amount of typical operation data, which is especially suitable for dynamically changing real - world scenarios.

Based on this, this research, using the 3D diffusion policy framework as the foundation, conducts in - depth research on how to utilize the spatial information collected by depth cameras to construct an efficient data collection process and improves the learning efficiency of robots for complex tasks by

optimizing policy training algorithms. The aim is to construct a complete chain of "spatial sensing - data acquisition - strategy training - deployment and execution", thereby enabling robots to perform manipulation tasks efficiently and stably in unstructured daily scenarios. This study will focus on the typical task of autonomous desktop clearing by robotic arms, verifying the effectiveness and practicality of the 3D diffusion policy framework through real - world scenarios, and providing theoretical and practical support for the application of Diffusion Policy in more complex scenarios.

II. Literature Review

A. Diffusion Model in Robotics

Generative models known as diffusion models have demonstrated remarkable capabilities in synthesizing high-quality images by gradually converting random noise into structured data samples [1] [2] [3] [4] . Building upon their exceptional representational power, these models have recently found applications across various robotics domains, including reinforcement learning [6] [7], imitation learning [5] [8] [9] [10], reward function learning [11] [12], object grasping [13] [14] [15], and trajectory generation [16] [17].

This project is based on the 3D Diffusion Policy framework [18], which introduces 3D point clouds into diffusion policy. Compared to the Diffusion Policy [5], which uses RGB images from two viewpoints as observations, the introduction of point cloud allows the model to better perceive the location of objects in 3D space, which significantly reduces the data required for training and improves the success rate of completing the task.

III. Method

Before introducing the actually steps of the process we will first go over the Key equations. The stages of our project could be divided into 3 parts: data collection, data preprocessing, training with diffusion policy, and real robot deployment.

A. Key equations

Just like the importance of velocity propagation and Jacobians in robotic manipulation, these functions play a crucial role in the training process of diffusion policy.

1) Noise prediction equation:

$$\hat{\epsilon} = \epsilon_\theta(z_t, A_t^k, k)$$

This is the core noise prediction formula in Diffusion Policy, which estimates future noise based on the current latent encoding z_t , the noisy action sequence A_t^k at step k, and the timestep k.

2) Denoising Iteration Formula:

$$A_t^{k-1} = \alpha_k \left(A_t^k - \gamma_k \hat{\epsilon}\right) + \sigma_k \mathcal{N}(0, I)$$

This equation describes the iterative denoising process, where the final goal is to transform the noisy sequence A_t^k into a clean sequence A_t^0.Here,$N(0,1)$represents Gaussian noise, which serves as the initial noise in the denoising process.

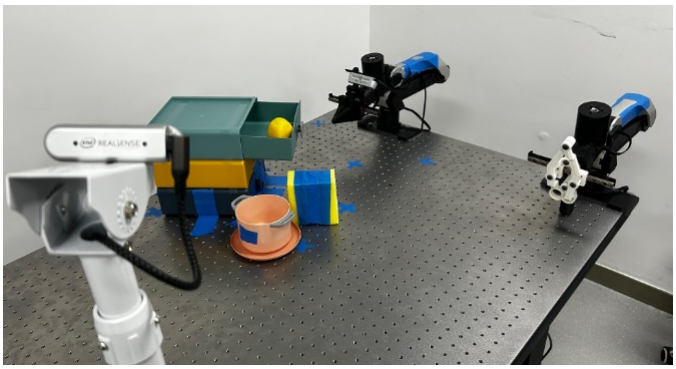


Fig. 1: Hardware setup

3) Discussion on Multi-Modal Capability: In real-world scenarios, there are often multiple valid solutions to a given task, rather than a single deterministic approach. However, traditional neural networks typically predict a single output, limiting their ability to handle tasks that require diverse strategies. In contrast, Diffusion Policy exhibits strong multi-modal performance.

A notable example is the "push T" task, where conventional models tend to favor a single direction (e.g., pushing a block either left or right). In contrast, Diffusion Policy can effectively explore and execute multiple feasible trajectories, such as pushing the block along either the left or right path, demonstrating its superior adaptability in real-world applications.

B. Data collection

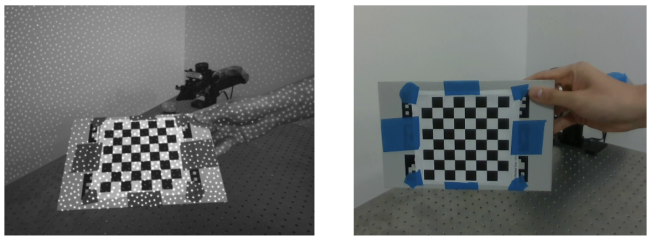
The main part of our data collection is that we collected the 'Action','Depth' and 'qpos' data that later on will be fed to the training and deployment process. We used Real-sense D435 as our camera for collecting depth images; ARX 5 as a Bilateral Teleoperation System for imitaion learning, and a couple of objects for the robot to grab and push.

We employ the Zarr file format for storing large-scale robotic demonstration data due to its chunk-based storage architecture, which offers significant advantages for machine learning pipelines. Its advantages lies in its high time efficiency, high scalability and high usage flexibility.

C. Camera Calibration

1) Intrinsic parameter–Zhang's ca. libration method :

1) **Checkerboard Preparation**:
   - Use an 8×8 grid with 15mm squares
2) **Multi-Angle Image Capture**:
   - Capture 15-20 images of the checkerboard from varying distances/angles
3) **Feature Detection**:
   - Detect checkerboard corners in 2D space
4) **3D-2D Mapping**:
   - Set $Z = 0$ on the XY plane
   - Compute precise 3D coordinates for all corners
5) **Parameter Estimation**:



IR_image RGB_image

Fig. 2: Calibration

- Solve for:
  - Intrinsic matrix (camera parameters):

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

  - Distortion coefficients (radial/tangential):

$$(k_1, k_2, p_1, p_2, k_3)$$

2) Extrinsic parameter–PNP method:

1) **Camera-to-Calibration Plate Transformation ($\mathbf{T}_{c\to w}$):**
   - Employ the Perspective-n-Point (PnP) method to compute the rotation matrix $\mathbf{R}_{c\to w}$ and translation vector $\mathbf{t}_{c\to w}$
   - The world coordinate system (w) is defined with its origin at the upper-left corner of the calibration plate
2) **Calibration Plate-to-Robot Base Transformation ($\mathbf{T}_{w\to b}$):**
   - Measure the physical transformation between coordinate systems:
     - Translation: -118 mm along x-axis and $+108$ mm along y-axis
     - Rotation: -90° clockwise rotation about the x-axis
   - The composite transformation is computed as:

$$\mathbf{T}_{w\to b} = \mathbf{T}_1 \times \mathbf{T}_2 \times \mathbf{T}_3$$

3) **Camera-to-Robot Base Transformation ($\mathbf{T}_{c\to b}$):**
   - Compute the final transformation through matrix multiplication:

$$\mathbf{T}_{c\to b} = \mathbf{T}_{c\to w} \times \mathbf{T}_{w\to b}$$

   - This yields the complete extrinsic calibration between camera and robotic arm base

The following is the result of our transformation matrix:

$$\mathbf{T}' \times (\text{Camera Coordinates}) = (\text{Robotic Arm Coordinates})$$

where $\mathbf{T}' = \mathbf{T}^{-1}$ is the inverse transformation matrix:

$$\mathbf{T}' = \begin{bmatrix} 0.75422983 & 0.23041472 & -0.6148548 & 468.233 \\ -0.65643515 & 0.28624051 & -0.69796795 & 1090.323 \\ 0.01517426 & 0.93004055 & 0.36714346 & -472.788 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Intrinsic Parameter Verification

- Perform quantitative comparison of reprojection errors across:
  - Manufacturer-provided specifications
  - Calibration-derived values
  - Theoretical calculations
- Focal length validation using field-of-view relationships:

$$f_x = \frac{\text{Image Width}}{2 \times \tan(\text{Horizontal FOV}/2)}$$
$$f_y = \frac{\text{Image Height}}{2 \times \tan(\text{Vertical FOV}/2)}$$

D. Calibration Parameter Verification

- Validate coordinate transformation between camera and world frames:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

- Quantitative assessment at calibration plate reference point:
  - Compare calculated vs. ground-truth coordinates
  - Special focus on upper-left corner point accuracy

The result of our calibration are as follows:

TABLE I: RGB Calibration Parameters Comparison

Parameter Type	Reprojection Error (pixels)
New Calibration Parameters	0.1699
Manufacturer Parameters	0.1943
Theoretical Parameters	0.5294

Parameter Set	Source	Win Rate
Intrinsic 1	Manufacturer	2/21 (9.52%)
Intrinsic 2	Calibrated	17/21 (80.95%)
Intrinsic 3	Theoretical	2/21 (9.52%)

Conclusion: Intrinsic 2 (Calibrated) performs best.

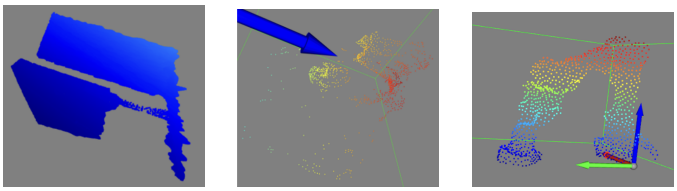
E. Preprocessing the depth image

1) Data Acquisition:

- Depth information acquired via `RealSense SDK` and published as ROS depth image messages
- Joint-state and depth data synchronized using `MessageFilter`
- Depth images converted to point clouds using `Open3D` with intrinsic/extrinsic camera parameters that is acquired through the calibration process.

2) Initial Challenges:

- Raw point clouds contain ∼200,000 points per frame with significant noise.
- Initial RANSAC plane segmentation approach proved computationally expensive:
  - Processing time: 5s per frame (non-real-time).
  - Limited effectiveness with `RealSense D435i`'s lower precision compared to L515 (this is mentioned in the paper).
  - Trade-off between preserving manipulator geometry and removing planar surfaces.



(a) Raw Point Cloud (b) RANSAC Filtering (c) Optimized Point Cloud

Fig. 3: Point cloud processing pipeline showing three stages: (a) unprocessed raw data containing approximately 200,000 points with significant noise and redundancy; (b) intermediate result after RANSAC-based plane segmentation for desk removal; (c) final optimized point cloud achieved through homogeneous transformation alignment followed by farthest point sampling (FPS) downsampling. The processing sequence demonstrates progressive improvement in point cloud quality while maintaining essential geometric features.

3) Optimized Preprocessing: The refined pipeline employs:

1) **Workspace Cropping**:
   - Axis-aligned bounding box filtering ($O(n)$ complexity)
   - Critical requirement: Proper alignment of transformation matrix's z-axis with physical workspace
2) **Farthest Point Sampling (FPS)**:
   - Downsample to 1,024 points using `Open3D` implementation
   - Preserves geometric features while reducing computational load

TABLE II: Processing Time Comparison

Method	Processing Time
RANSAC Segmentation	12000 ms
Optimized Pipeline	10-20 ms

4) Performance Metrics:

- Achieves real-time performance (10× faster than depth image publishing rate)
- Enables effective model inference while maintaining spatial awareness

F. Model Training

1) Dataset Construction: The task-specific dataset is structured in Zarr format with four data modalities:

- `depth`: Depth images from RealSense D435i
- `action`: 6-DOF end-effector actions (position + orientation)
- `point_cloud`: 3D point clouds (converted from depth images)
- `qpos`: 7-dimensional joint position vectors

Each dataset contains multiple episodes, where an episode comprises sequential steps. The number of episodes and steps per episode varies with task complexity (detailed in Section III-G).

2) Model Configuration: We implement a 3D Diffusion Policy with:

- Network architecture: same as the article with 3-layer MLP with increasing dimensions (64→128→256).
- Diffusion steps: 100 (linear noise schedule)
- Training: AdamW optimizer ($\eta = 1.0 \times 10^{-4}$), 1001 epochs

3) Robustness Enhancement: Through empirical analysis, we identify optimal control parameters:

$$\text{Planning horizon} = 16 \text{ steps}$$
$$\text{Action execution} = 8 \text{ steps}$$

This configuration improves motion smoothness (measured by jerk reduction through physical witness) and task success rates.

G. Policy Deployment and Experimental Validation

1) Deployment Pipeline:

- Trained models saved as `.ckpt` files
- Real-time inference on host PC (less than 200ms)
- Action commands published via ROS (`/arm_control` topic)

TABLE III: Task Complexity Classification

Category	Description	Metrics
Basic	Single primitive action	Success rate
Advanced	Multi-step manipulation	Sub-task completion

2) Task Taxonomy:

a) Basic Tasks:

- **Bowl Task**: Grasp plastic bowl from table (tests spatial perception)
- **Drawer Task**: Close opened drawer (tests constrained motion learning)

b) Advanced Tasks:

- **Lemon-Bowl Task**: Sequential pick-and-place (lemon → bowl)
- **Lemon-Drawer Task**: Triple-action sequence (grasp → place → close)

The advanced tasks evaluate the policy's ability to:

- Maintain object permanence
- Handle compound actions
- Manage state transitions

IV. Result

We tested the performance of the diffusion model on the Bowl task, the Drawer task, the Lemon_Bowl task and the Lemon_Drawer task. The number of rounds (episodes) and the number of steps per round for the corresponding dataset are shown in the table IV:

TABLE IV: Dataset Parameter

Parameter Name	**Task Name**			
	Bowl	***Drawer***	***Lemon_Bowl***	***Lemon_Drawer***
Episode Number	20	20	40	40
Episode Length	150	100	300	400

We measured the success rate and completion time for the different tasks: for each task, twenty experiments were performed, the success rate was calculated by the number of successes/total number of experiments, and the completion time was the average completion time of all successful attempts, as shown in the table V:

TABLE V: Success Rate and Completion Time

Parameter Name	**Task Name**			
	Bowl	***Drawer***	***Lemon_Bowl***	***Lemon_Drawer***
Success Rate	70%	100%	30%	10%
Completion Time	14.64s	11.14s	55.62s	65.71s

The experiments introduced a certain amount of randomness. In the Bowl experiment, the position of the bowl placement as well as its orientation was randomly distributed over a range. In the Drawer experiment, the degree of drawer opening was randomly distributed. In the Lemon_Bowl experiment, the positions of the lemon and the bowl placement were randomly distributed over a range. And the position of the lemon and the degree of drawer opening in the Lemon_Drawer experiment were randomly distributed. This is designed to test the algorithm's generalizability.

V. Discussion

The algorithm demonstrates superior performance in simpler tasks such as Bowl and Drawer. When handling complex tasks like Lemon_Bowl and Lemon_Drawer, it exhibits a significant decline in success rate and a prolonged completion time.

The limited capability in complex tasks stems from multiple factors. First, our system faces hardware limitations: specifically, we employed the D435i camera, which was not recommended by the original authors due to its low accuracy. This camera may generate point clouds with insufficient precision to facilitate complex task completion. Second, constrained by inadequate training data and resource limitations during data acquisition, the model's performance on complex tasks could potentially be improved through expanded dataset collection and rigorous preprocessing.

During experimental evaluation, the imitation learning algorithm demonstrated notable adaptability and stability. For instance, when initial grasping attempts fail, the algorithm autonomously initiates subsequent attempts until successful object acquisition occurs. This behavior arises because the system computes actions based on observed point clouds; when these indicate unsuccessful grasping, the robotic arm automatically repeats the attempt. Similarly, if the target object's position changes unexpectedly during task execution, the algorithm promptly directs the robotic arm to locate and pursue the new position. This dynamic environment adaptability and stability represent significant advantages over traditional planning-based algorithms.

VI. Conclusion

This project focuses on robotic arm operations in complex daily scenarios, building upon an existing diffusion strategy framework to establish a comprehensive technical system encompassing data collection, policy training, and deployment execution. The system utilizes depth cameras to acquire scene depth information and incorporates innovative point cloud processing technology, enabling accurate spatial perception without relying on external positioning markers. Our independently developed teleoperation data collection program provides substantial training data for model development. Furthermore, the integration of model predictive control principles has significantly improved policy robustness in complex environments.

Experimental results demonstrate the system's strong performance across multiple scenarios, exhibiting both high training efficiency and exceptional generalization capabilities. Throughout the project, we have accumulated valuable practical experience in several key areas: operating physical robotic arms, utilizing depth cameras effectively, mastering point cloud processing techniques, and gaining hands-on expertise in imitation learning applications.

Acknowledgment

We sincerely thank Professor Qinhu Meng and Professor Junwei Liu for their invaluable guidance, critical feedback, and continuous support throughout this research. Their expertise profoundly shaped the direction of this work, and we are deeply grateful for their mentorship.

We extend our gratitude to Yue Hong, Lingxiao Meng, Hao Chen, Menglian Li, Tongxue Yue for their technical assistance and constructive suggestions during experiments. Their dedication significantly facilitated our progress.

We acknowledge Department of Electronic and Electrical Engineering and School of System Design and Intelligent Manufacturing for providing essential resources and computational infrastructure.

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. NeurIPS, 2020

[2] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. ICLR, 2021

[3] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bj¨orn Ommer. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022

[4] Jiaming Song, Chenlin Meng, and Stefano Ermon. De- noising diffusion implicit models. ICLR, 2021.

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

[6] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? arXiv preprint arXiv:2211.15657, 2022.

[7] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. ICLR, 2023

[8] Tim Pearce, Tabish Rashid, Anssi Kanervisto, Dave Bignell, Mingfei Sun, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Ida Momennejad, Katja Hofmann, et al. Imitating human behaviour with diffusion models. ICLR, 2023.

[9] Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. arXiv preprint arXiv:2304.02532, 2023.

[10] Zhou Xian, Nikolaos Gkanatsios, Theophile Gervet, Tsung-Wei Ke, and Katerina Fragkiadaki. Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation. CoRL, 2023.

[11] Tao Huang, Guangqi Jiang, Yanjie Ze, and Huazhe Xu. Diffusion reward: Learning rewards via conditional video diffusion. arXiv, 2023

[12] Felipe Nuti, Tim Franzmeyer, and Jo˜ao F Henriques. Extracting reward functions from diffusion models. arXiv preprint arXiv:2306.01804, 2023.

[13] Tianhao Wu, Mingdong Wu, Jiyao Zhang, Yunchong Gan, and Hao Dong. Learning score-based grasping primitive for human-assisting dexterous grasping. In NeurIPS, 2023.

[14] Julen Urain, Niklas Funk, Jan Peters, and Georgia Chal-vatzaki. Se (3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. In 2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023

[15] Anthony Simeonov, Ankit Goyal, Lucas Manuelli, Lin Yen-Chen, Alina Sarmiento, Alberto Rodriguez, Pulkit Agrawal, and Dieter Fox. Shelving, stacking, hanging: Relational pose diffusion for multi-modal rearrangement. arXiv preprint arXiv:2307.04751, 2023

[16] Kallol Saha, Vishal Mandadi, Jayaram Reddy, Ajit Srikanth, Aditya Agarwal, Bipasha Sen, Arun Singh, and Madhava Krishna. Edmp: Ensemble-of-costs-guided diffusion for motion planning. arXiv, 202

[17] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. arXiv, 2022.

[18] Ze Yanjie, et al. “3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations.” ArXiv.org, 2024, arxiv.org/abs/2403.03954.