Learning from Offline Human Demonstrations with Diffusion Policy and Flow Matching

Sandeep Zachariah
Carnegie Mellon University
szachar@andrew.cmu.edu

Hayden Feddock
Carnegie Mellon University
hfeddock@andrew.cmu.edu

Teddy Lin
Carnegie Mellon University
eplin@andrew.cmu.edu

Zhewen Zheng
Carnegie Mellon University
zhewenz@andrew.cmu.edu

Joshua Momo
Carnegie Mellon University
jmomo@andrew.cmu.edu

Abstract: Offline learning from human demonstrations offers a scalable alternative to online robot training, but non-Markovian behavior, heterogeneous data quality, and sensitivity to design choices complicate reliable policy learning. Building on the RoboMimic benchmark [1], we evaluate diffusion-based and flow-matching action-sequence policies under a unified protocol and compare directly against the reported baselines. Diffusion policies model multimodal action distributions via conditional denoising, while flow matching learns deterministic continuous-time transports that enable efficient sampling. Beyond reproducing these methods, we study the impact of key design choices, including goal conditioning and observation horizon length. Using the original datasets and evaluation metrics, we show that diffusion policies achieve state-of-the-art performance on long-horizon manipulation tasks, while flow matching provides a competitive and efficient alternative.

Keywords: Offline Learning, Diffusion Models, Flow Matching

1 Introduction

Offline learning from human-labeled data originally gained popularity in computer vision, but in the last decade it has also emerged as a compelling paradigm for robot learning. Robots can acquire complex manipulation skills directly from previously collected demonstrations, without requiring costly or unsafe online interaction [2]. This paradigm offers several advantages: *safety*, since policies never explore in the real world during training; *speed*, since learning can proceed faster than real time; *reproducibility*, due to fixed datasets; and *data availability*, enabling the reuse of large human demonstration corpora.

RoboMimic [1] standardized offline manipulation learning and made clear that success hinges on modeling temporal dependencies, handling demonstration noise/heterogeneity, and making careful architecture and training choices with surrogate imitation losses often poorly predicting task success. Since then, generative action-sequence policies such as Diffusion Policy [3] and more recent Flow Matching approaches [4, 5] have reported strong results in manipulation, but these methods are typically evaluated under differing training budgets, implementations, and design choices, making it difficult to attribute performance gains or compare methods fairly. This work revisits RoboMimic under a controlled protocol: we reproduce diffusion and flow-matching policies on the proficient-human (PH) datasets, compare directly against the original RoboMimic baselines, and run targeted

ablations (goal conditioning, observation horizon) to isolate which design decisions materially affect success.

The specific contributions of this project include:

1. **Unified evaluation of generative policies.** We evaluate diffusion and flow-matching policies on RoboMimic PH tasks under a consistent training and evaluation protocol.
2. **Direct comparison to RoboMimic baselines.** We report results alongside the baselines from RoboMimic to enable a controlled, task-matched comparison.
3. **Targeted ablations.** We study goal conditioning and observation horizon to identify which design decisions drive (or hinder) performance.

Our results provide a modernized comparison against the original RoboMimic baselines and offer insights into how generative action models address the long-standing challenges identified in offline robot learning.

2 Related Work

Offline learning has been widely studied for robot manipulation as a means of acquiring skills from fixed demonstration datasets without online interaction. Early approaches primarily relied on supervised imitation, such as Behavioral Cloning (BC) and recurrent extensions to address partial observability, but these methods are sensitive to covariate shift, dataset quality, and multimodal human behavior. Offline reinforcement learning methods, including BCQ, CQL, and IRIS, introduced policy constraints and regularization to mitigate distributional mismatch, yet often required careful tuning and struggled in complex, high-dimensional manipulation tasks.

The RoboMimic benchmark [1] provided a standardized evaluation suite for offline manipulation learning and highlighted persistent challenges such as non-Markovian demonstrations, heterogeneous data quality, and weak correlation between training loss and task success. These findings motivated the development of policy representations capable of modeling temporal structure and multimodal action distributions while remaining stable under noisy supervision.

Generative sequence models have recently emerged as a promising direction. Diffusion Policy [3] models action generation as a conditional denoising process over short horizons, demonstrating strong stability and improved performance across manipulation benchmarks. Flow matching offers an alternative formulation that learns a deterministic continuous-time mapping from noise to expert actions, enabling faster inference with competitive performance [5]. The present work builds on these approaches by directly comparing diffusion and flow-matching policies within the RoboMimic framework and analyzing the effect of goal conditioning and observation horizon under consistent experimental settings.

3 Methodology

3.1 Diffusion Models and Diffusion Policies

3.1.1 Diffusion Models

Diffusion models are generative models that learn to synthesize data by inverting a gradual noising process. A *forward* Markov chain progressively corrupts a clean sample $x_0 \sim q_{\text{data}}$ with Gaussian noise,

$$q(x_t \mid x_{t-1}) = \mathcal{N}\big(x_t; \sqrt{1-\beta_t}\, x_{t-1},\, \beta_t I\big), \tag{1}$$

until it becomes nearly pure noise x_L. The top row of Figure 1 illustrates this process for images. A neural network is then trained to approximate the *reverse* denoising dynamics by predicting the noise added at each step, using the simple mean-squared error loss

$$\mathcal{L}(\theta) = \mathbb{E}_{x_0,\varepsilon,t}\big[\|\varepsilon - \varepsilon_\theta(x_t, t)\|_2^2\big], \tag{2}$$

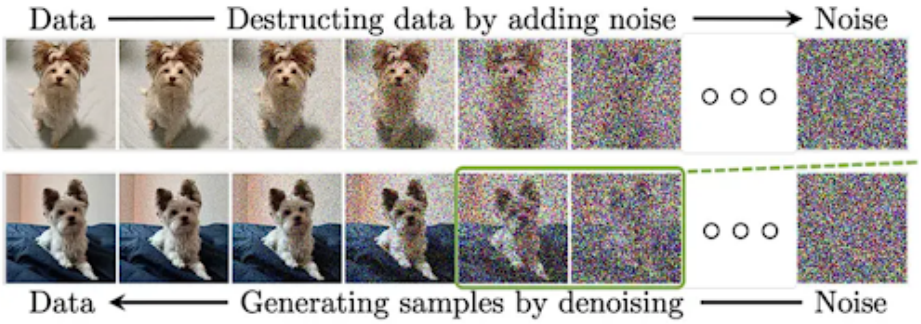

Figure 1: Illustration of the diffusion process. *Top:* forward process that corrupts data into noise. *Bottom:* learned reverse process that iteratively denoises noise back into data.

so that starting from $x_L \sim \mathcal{N}(0, I)$ and iteratively denoising recovers a clean sample x_0 (bottom row of Figure 1).

3.1.2 Diffusion Policy for Robot Control

Diffusion Policy adapt this idea from images to robot action sequences. Instead of pixels, the data are short "action chunks" $a_{t:t+H_{\text{act}}-1}$ extracted from demonstrations, conditioned on an observation history $o_{t-H_{\text{obs}}+1:t}$ and an optional goal g. The policy models

$$p_\theta\big(a_{t:t+H_{\text{act}}-1} \mid o_{t-H_{\text{obs}}+1:t}, g\big) \tag{3}$$

via a conditional diffusion process on actions: expert action chunks are noised forward as above, and a U-Net-based denoising network ε_θ is trained to predict the noise given the noised actions, timestep, and an embedding of $(o_{t-H_{\text{obs}}+1:t}, g)$.

At test time, the policy repeatedly samples an action chunk by running the reverse diffusion process conditioned on the most recent observation history, but only executes the first action before replanning at the next step. This receding-horizon strategy yields smooth, temporally coherent actions while retaining closed-loop feedback, and has been shown to outperform standard behavior cloning on RoboMimic-style manipulation tasks.

3.2 Flow Matching Policy

Flow matching offers an alternative generative modeling approach that directly learns a deterministic continuous-time vector field transporting a noise distribution to the data distribution. Rather than learning to reverse a multi-step diffusion chain, flow-matching models learn the instantaneous velocity that moves a sample along a smooth trajectory from noise to expert actions.

For robot control, our flow matching policy $\pi(a \mid o)$ is implemented as a **Conditional U-Net 1D** conditioned on observation history and trained to generate short-horizon action sequences via the flow-matching ordinary differential equation (ODE).

Our flow matching policy uses an architecture adapted from Diffusion Policy [3]. The network architecture consists of:

- **Input:** Noisy action chunk $\mathbf{a}_t \in \mathbb{R}^{H \times d_a}$ (linearly interpolated between noise and target)
- **Observation Encoder:** For low-dimensional observations $\mathbf{o} \in \mathbb{R}^{T_o \times d_o}$, we use a 3-layer MLP with hidden dimensions [256, 512, 256], ReLU activations, and LayerNorm. The two observation timesteps are flattened and concatenated before encoding, producing $\mathbf{e}_o \in \mathbb{R}^{256}$.
- **Time Embedding:** Sinusoidal positional encoding of flow time $t \in [0, 1]$, projected to $\mathbf{e}_t \in \mathbb{R}^{128}$.

- **U-Net Backbone:** 1D convolutional encoder-decoder with skip connections:
  - Encoder: 4 downsampling blocks with Conv1D (kernel size 5, stride 2), GroupNorm, and SiLU activation
  - Bottleneck: 2 residual blocks with attention mechanisms
  - Decoder: 4 upsampling blocks with transposed Conv1D and skip connections
  - Channel dimensions: [256, 512, 1024, 2048] at each level
- **Output:** Predicted vector field $\mathbf{v}_\theta(t, \mathbf{a}_t, \mathbf{o}) \in \mathbb{R}^{H \times d_a}$

Compared to diffusion policies, flow matching offers faster sampling, due to its single-step ODE integration, competitive generative expressiveness, and comparable stability in high-dimensional control settings [4]. These advantages make flow matching a promising alternative for real-time manipulation.

3.3 Diffusion Extension: Goal Conditioning

In our setting, goal conditioning is implemented by providing the diffusion policy with an explicit representation of the desired *object* configuration, rather than the full robot state. Let $o_{t-H_{\text{obs}}+1:t}$ denote the recent observation history (including robot kinematics and object state), and let g_{obj} denote the final desired object state extracted from the demonstration (e.g., the object's pose in the last timestep). The goal-conditioned diffusion policy predicts an action sequence as

$$\hat{a}_{1:H_{\text{act}}} = \pi_\theta\big(o_{t-H_{\text{obs}}+1:t},\, g_{\text{obj}}\big), \tag{4}$$

allowing the model to anchor its predictions to the target object arrangement without requiring a full goal specification for the robot arms themselves. During training, g_{obj} is concatenated with the encoded observation history and injected as part of the conditioning signal for the denoising network.

At inference time, the policy follows the standard receding-horizon procedure: starting from Gaussian noise, the model iteratively denoises an action sequence while being conditioned jointly on the current observation window and the object-goal vector. The resulting clean action sequence,

$$x^{(0)} = f_\theta\big(x^{(L)},\, o_{t-H_{\text{obs}}+1:t},\, g_{\text{obj}}\big), \tag{5}$$

provides a short-horizon plan whose first action is executed before replanning. Conditioning on the final object state helps reduce ambiguity arising from multimodal human demonstrations and enforces trajectory predictions that are consistent with the intended task completion.

3.4 Diffusion Extension: Horizon Ablation

Diffusion Policy conditions its predicted action sequence on a finite window of past observations, referred to as the *observation horizon*. Let

$$o_t \in \mathbb{R}^{d_o}$$

denote the low-dimensional observation at time step t (e.g., joint positions, end-effector pose, gripper state). For an observation horizon of length H_{obs}, the policy at time t is conditioned on the stacked history

$$\mathbf{o}_{t-H_{\text{obs}}+1:t} = \big(o_{t-H_{\text{obs}}+1}, \ldots, o_{t-1}, o_t\big),$$

which is encoded to produce the conditioning vector for the diffusion model. Increasing H_{obs} injects more temporal context into the policy but also increases the dimensionality of the conditioning input and the difficulty of optimization. Intuitively, very small horizons ($H_{\text{obs}} = 1$ or 2) provide too little history, while very large horizons add mostly redundant information and computational cost. We therefore expect a "sweet spot" at a moderate observation horizon that captures the short-term dynamics relevant for control without incurring unnecessary overhead.

4 Experimental Results

All policies are trained and evaluated within the standard RoboMimic low-dimensional observation setting, using the official benchmark tasks (Lift, Can, Square, Transport, and Tool Hang). To enable a fair comparison, we follow the original Diffusion Policy implementation and match its key hyperparameters, including a batch size of 256 and DDIM inference with 16 sampling steps. Both Diffusion Policy (DP) and Flow Matching (FM) are trained using a unified training pipeline and identical observation/action interfaces.

Training duration varies only due to practical runtime considerations. Models are trained for 2000 epochs on Lift and Can, which have relatively fast environment rollouts, and for 1000 epochs on Square, Transport, and Tool Hang, whose longer episode lengths make training substantially more time-consuming. Our evaluation focuses on the CNN-based Diffusion Policy; Transformer-based variants are not included due to scope. All results report success rates computed over 50 rollouts per task, with no online interaction or environment modifications during training or evaluation.

	BC	BC-RNN	BCQ	CQL	HBC	IRIS	DP	FM
Lift	**100.0**	**100.0**	**100.0**	92.7	**100.0**	**100.0**	**100.0**	**100.0**
Can	95.3	**100.0**	88.7	38.0	**100.0**	**100.0**	**100.0**	**100.0**
Square	78.7	**84.0**	50.0	5.3	82.6	78.7	58.0	**84.0**
Transport	17.3	71.3	7.3	0.0	48.6	41.3	**76.0**	64.0
Tool Hang	29.3	19.3	0.0	0.0	30.0	11.3	**86.0**	70.0

Table 1: Maximum achieved Average Returns with low_dim Proficient-Human (PH)

The CNN-based Diffusion Policy (DP) produced results that are broadly consistent with prior work, while achieving substantially stronger performance on long-horizon manipulation tasks under our training protocol. On Square, DP achieves 58% success, underperforming BC-RNN (84.0%) and HBC (82.6%), but still outperforming BCQ (50.0%) and CQL (5.3%). In contrast, DP demonstrates clear advantages on long-horizon, contact-rich tasks: on Transport, DP reaches 76.0% success, outperforming all baseline methods and exceeding BC-RNN (71.3%); on Tool Hang, DP achieves 86.0%, likewise surpassing every baseline by a wide margin. These results indicate that diffusion-based action-sequence modeling is particularly effective for tasks requiring multi-stage coordination and sustained contact.

The lower performance on Square, relative to the 100% success reported in [3], likely reflects differences in training budget and optimization settings. Square requires precise, fine-grained corrective actions during alignment, making it especially sensitive to training duration and gradient variance. Under reduced training budgets, these sensitivities appear to disproportionately affect diffusion-based policies.

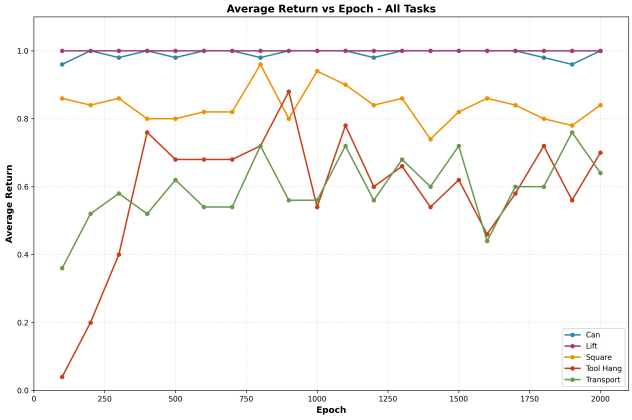

Figure 2: Performance of flow matching model during training across task sets.

Flow Matching (FM) also achieves strong performance across the benchmark, matching or exceeding most classical offline RL baselines. On Transport, FM attains 64.0% success, substantially outperforming BC, BCQ, and CQL, though remaining below DP and BC-RNN. FM performs competitively on Square, matching BC-RNN as the highest-performing method, and achieves 70.0% on Tool Hang. While FM does not surpass the best-performing diffusion policy on the most difficult long-horizon tasks, its consistently strong results across tasks suggest that flow matching is a viable and efficient alternative to diffusion for offline manipulation learning.

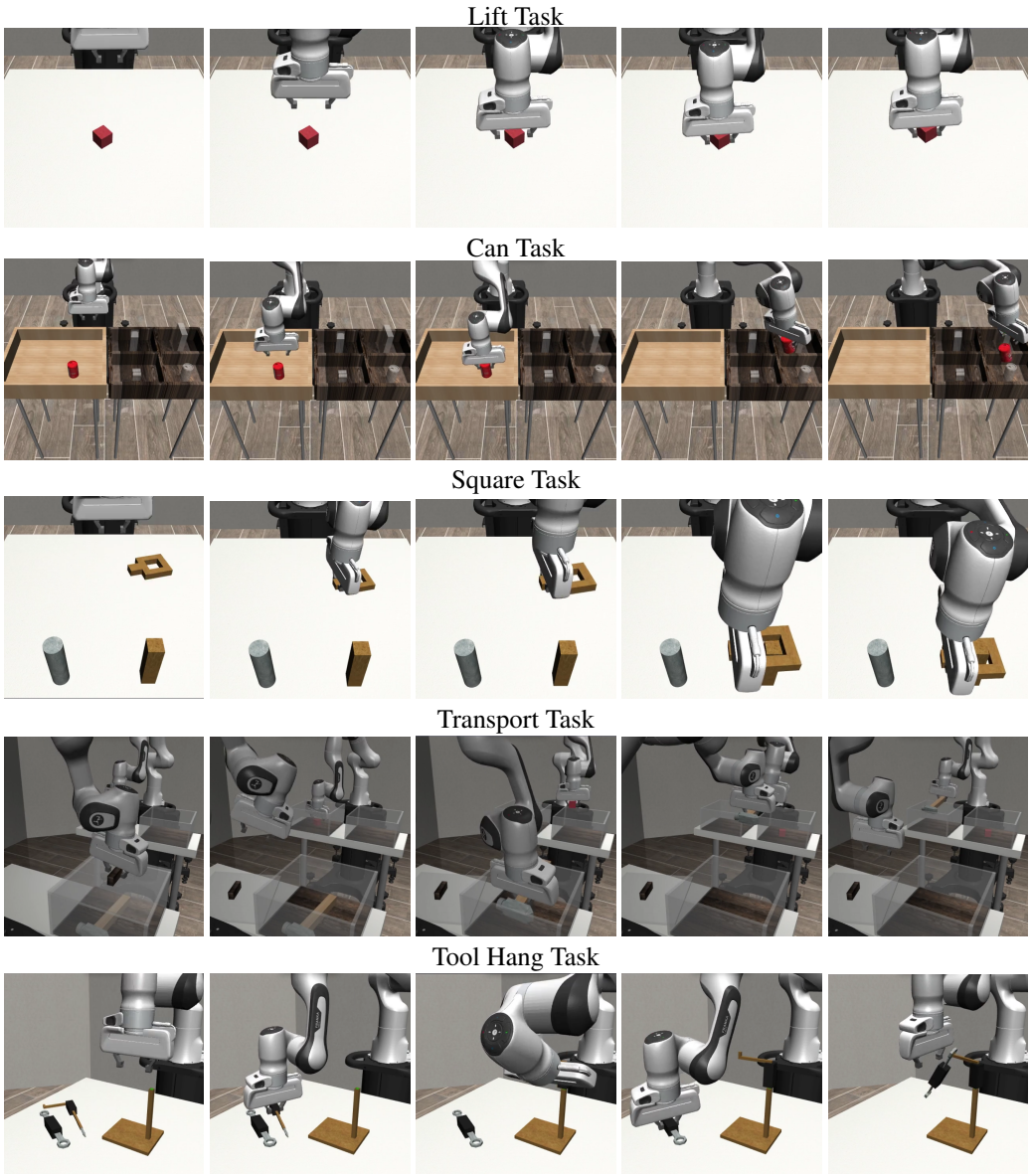

Figure 3: Flow Matching Rollout visualizations for each task in the set.

4.1 Ablation Study I: Goal Conditioning

We study the effect of explicit goal conditioning on low-dimensional Diffusion Policies across all five RoboMimic tasks. The goal is defined as the final object state from each demonstration, excluding robot arm state. Since evaluation rollouts do not provide a ground-truth goal, we condition the policy at inference on a fixed global goal vector, computed as the mean of final object states over

the training set. All models share the same dataset, architecture, optimizer, and diffusion hyperparameters, and are trained for 600 epochs.

Across tasks, goal-conditioned policies consistently underperform the standard Diffusion Policy baseline on pick-and-place–style tasks (Can, Square, Transport), and only partially recover performance on Tool Hang (Table 2). This degradation is likely due to a mismatch between demonstration-specific goals and the fixed global goal used at inference, which weakens the conditioning signal. In addition, increasing the conditioning dimensionality may hinder optimization when the object-only goal provides limited information relative to trajectory-level contact dynamics. These results suggest that naïve object-only goal conditioning is insufficient in this setting and may degrade performance without more informative or task-specific goal representations.

	DP	**DP + Goal**
Lift	**100.0**	**100.0**
Can	**100.0**	0.0
Square	58.0	12.0
Transport	76.0	0.0
Tool Hang	86.0	43.3

Table 2: Success rate (%) with and without goal conditioning on low-dimensional PH datasets.

4.2 Ablation Study II: Observation Horizon

We study the effect of the observation horizon on policy performance using a low-dimensional Diffusion Policy on the `Square` task. All experiments use the same dataset, architecture, optimizer, training schedule, action horizon H_{act}, and prediction horizon H_{pred}; only the observation horizon is varied.

We sweep

$$H_{\text{obs}} \in \{1, 2, 4, 6\},$$

training each model from scratch under identical settings. Policies are evaluated every 50 epochs over a fixed number of rollouts with a maximum episode length of 400 steps. As shown in Fig. 4, performance peaks at small horizons ($H_{\text{obs}} = 2$) and degrades for larger values. This indicates that a short observation history is sufficient for Square, while longer horizons increase conditioning dimensionality and optimization difficulty without providing additional benefit.

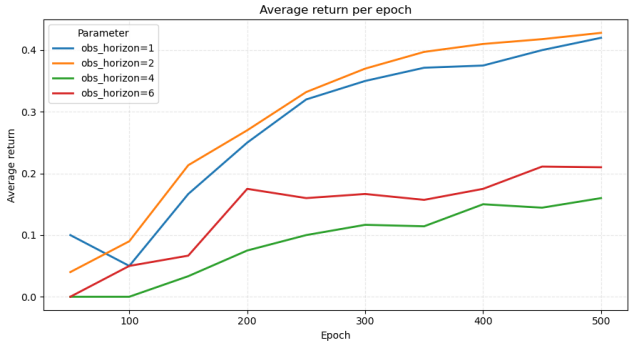

Figure 4: Average Return of Diffusion Policy for different values of *observation_horizon* on the *Square* task.

5 Conclusion

Offline learning from human demonstrations offers a scalable and safe path to robot skill acquisition, but prior work such as RoboMimic [1] has shown that non-Markovian behavior, heterogeneous

data quality, and sensitivity to design choices make reliable policy learning difficult. Building on diffusion policies [3] and flow-matching approaches [4, 5], we treated these methods as expressive generative sequence models for robot actions and evaluated them under a unified protocol on the RoboMimic proficient-human benchmark. Our experiments compared Diffusion Policy and Flow Matching against the original baselines (BC, BC-RNN, BCQ, CQL, HBC, IRIS) using the same low-dimensional observation setting and consistent training and evaluation pipelines. Both generative methods achieved strong performance on long-horizon, contact-rich tasks such as Transport and Tool Hang, and were competitive with BC-RNN on easier tasks, while Flow Matching additionally provided faster inference with comparable or better success rates.

At the same time, our results highlight that these models are highly sensitive to architectural and training choices. Diffusion Policy underperformed prior reports on Square, suggesting that factors such as training duration and batch size can materially affect final success. Our ablations showed that naïve object-only goal conditioning with a global mean goal vector degraded performance relative to the standard diffusion baseline, indicating that simple goal parameterizations may introduce harmful mismatches between training and evaluation. Likewise, increasing the observation horizon beyond a short history did not help on Square and instead reduced returns, likely due to higher-dimensional conditioning and more difficult optimization without additional useful information. Together, these findings suggest that while diffusion and flow-matching policies are promising tools for offline robot learning, realizing their full potential will require more careful design of goal representations, temporal context, and training protocols, as well as broader evaluations in image-based and real-robot settings.

Acknowledgments

Thank you to the developers of the RoboMimic and Robosuite packages.

References

[1] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation, 2021. URL https://arxiv.org/abs/2108.03298.

[2] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation, 2018. URL https://arxiv.org/abs/1806.10293.

[3] 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, 2024. URL https://arxiv.org/abs/2303.04137.

[4] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

[5] F. Zhang and M. Gienger. Affordance-based robot manipulation with flow matching, 2025. URL https://arxiv.org/abs/2409.01083.