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

Flow Before Imitation: Learning Dexterous In-hand Manipulation with Dynamic Visuotactile Shortcut Policy

Yijin Chen[1*], Wenqiang Xu[1*], Zhenjun Yu[1], Tutian Tang[1], Yutong Li[1], Siqiong Yao[1], Cewu Lu[1]

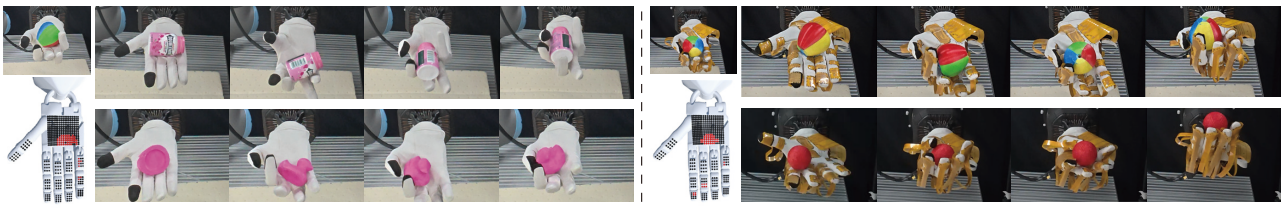


Fig. 1: We propose **Flow Before Imitation (FBI)**, a novel dynamic visuotactile imitation learning algorithm for dexterous in-hand manipulation. FBI's design enables two operational modes: with or without physical tactile sensors in the real world, largely extending the application scenarios.

***Abstract*— Dexterous in-hand manipulation remains a long-standing challenge in robotics, primarily due to the complex contact dynamics and partial observability. While humans synergize vision and touch for such tasks, robotic approaches often prioritize one modality, therefore limiting adaptability. This paper introduces Flow Before Imitation (FBI), a visuo-tactile imitation learning framework that dynamically fuses tactile interactions with visual observations through motion dynamics. Unlike prior static fusion methods, FBI establishes a causal link between tactile signals and object motion via a dynamics-aware latent model. FBI employs a transformer-based interaction module to fuse flow-derived tactile features with visual inputs, training a one-step diffusion policy for real-time execution. Extensive experiments demonstrate that the proposed method outperforms the baseline methods in both simulation and the real world on two customized in-hand manipulation tasks and three standard dexterous manipulation tasks. Code, models, and more results are available on the website https://sites.google.com/view/dex-fbi.**

I. Introduction

In-hand manipulation, which aims to reposition objects within a single dexterous hand, remains a critical, yet unsolved, problem in robot learning due to the complex contact dynamics and partial observability. Humans can easily interact with objects in their hands thanks to the fusion of vision and touch. Vision tracks global object states for task completion, while tactile sensing enables precise force adjustments during contact. Existing robotic approaches often prioritize either vision [1]–[3] or tactile sensing [4]–[6], limiting their effectiveness in complex manipulation scenarios. While recent visuotactile approaches [5], [7]–[10] demonstrate improved manipulation stability, most rely on optical tactile sensors [5], [7]–[9] whose bulkiness (typically $\geq 8mm$ thickness) prevents deployment across full-hand articulations (*e.g.*, metacarpal joints). Distributed tactile arrays (*e.g.*, piezoresistive arrays) present a practical alternative [4], [11], [12], offering hardware compatibility with commercial dexterous manipulators, full-palm coverage, and sufficient contact force resolution for detecting slip. Despite these advantages, the fusion of distributed tactile data with visual streams remains understudied in policy learning frameworks.

Existing visuotactile fusion methods typically process multimodal inputs through *static fusion*, such as introducing encoded features [9] or augmenting visual point clouds with tactile contact coordinates [10] from a single frame. These approaches neglect the intrinsic causal relationship between tactile interactions and object state transitions during dynamic manipulation: Tactile forces drive object state changes. Therefore, we take the path of *dynamic fusion*, extracting tactile information from temporal object motion flow via a dynamics-aware latent model. This representation enables two operational modes in practice: (1) Vision-Only mode. We can infer tactile cues without physical sensors; (2) Visuo-tactile mode. With tactile sensors, measured contact forces can be used to refine flow predictions, thereby improving precision. Therefore, our method can be flexibly deployed to a wider range of application scenarios, even when the tactile sensor is unavailable (Figure 1).

Building on this dynamic perspective, we present Flow Before Imitation (FBI) — an imitation learning framework integrating tactile and visual cues through motion dynamics. FBI processes multimodal inputs, including consecutive robot states, partially observed point clouds, and tactile readings, through dedicated encoder networks to extract proprioception and contact features. These encoded features are then dynamically fused and conditioned into a one-step shortcut model [13] to predict action sequences. This design enables efficient, real-time policy execution while integrating tactile-visual dynamics for robust in-hand manipulation.

To evaluate our method, we test FBI on five tasks, including in-hand reorientation, in-hand pushing, and three dexterous tasks from the public benchmark Adroit [14], which involve objects ranging from standard simple cubes to real-world items. Compared with four baseline methods including DP3 [15], ManiCM [16], Ada-Flow [17], and Consistency Policy [18], the proposed method performs better. In simulations, it reaches 64.7% (Vision-Only) to 66.5% (Visuotactile) average success, 16.6% to 18.4% higher

[1]Shanghai Jiao Tong University. * indicates equal contribution.

than the previous SOTA method, respectively. It is especially effective at hard reorientation tasks, with improvements of 19.3% (Vision-Only) to 21.4% (Visuotactile) higher. We also conduct real-world tests that show similar gains, ranging from 33.5% (Vision-Only) to 35.0% (Visuotactile) compared to an 18.5% baseline.

Our contribution can be summarized as follows:

1) We propose Flow Before Imitation (FBI), a visuotactile fusion method that dynamically infers tactile interactions from object motion flow, enabling real-time control via a one-step diffusion policy. It enables two operational modes, with or without physical tactile sensors, in real-world experiments, thereby largely extending the application scenarios.

2) Extensive evaluation across five tasks in both simulation and real-world experimental settings demonstrates FBI's superiority over baseline methods.

II. Related Works

A. Multimodal Sensing for Dexterous Manipulation.

Considering that humans operate with dexterous manipulation, multimodal sensing, especially vision and touch, is of concern. To empower robots with such abilities, researchers often start with simplified, single-modal perception settings, including vision-only [1]–[3], [19] and touch-only [4]–[6]. Though much progress has been made, vision-only methods, by nature, struggle with occlusions, while tactile-only systems lack global spatial awareness. Recent advances highlight the benefits of multimodal sensing. Guzey *et al*. [8], [9] combined self-supervised tactile pre-training with visual inputs to address contact reasoning, outperforming single-modal baselines. Yuan *et al*. [10] introduced a point cloud-based tactile representation fused with vision to enhance spatial planning. While these methods demonstrate the potential of multimodal fusion, some works exclusively use finger contacts as effectors while neglecting palm interactions [7], [9], and others restrict object rotation to predefined axes [4], [10]. In contrast, our approach utilizes the whole palm side to sense and manipulate objects, enabling the ability to reorient irregular objects and execute controlled pushes.

B. Imitation Learning for Dexterous Manipulation.

Since analytical planning [20] or control [21] has limited generalization ability to object variance, learning for dexterous manipulation gains increasing attention. While reinforcement learning (RL) [3], [5], [7] excels without demonstrations, it requires intricate reward design. On the contrary, imitation learning methods can mitigate the reward design problem and are known to have better sample efficiency with demonstration data. The policy learning methods in imitation are developed along with the generative methods, from nearest neighbors-based approaches [8], [9], [19], Gaussian-based dynamic motion primitives [22], generative adversarial networks [23], to diffusion models [15]–[18]. Ze *et al*. [15] replaced 2D image inputs in vanilla diffusion policy [24] with 3D point clouds, which significantly enhanced the performance of dexterous manipulation. Later, some works [17], [25] have replaced the diffusion process with flow matching [13], [26] to reduce inference time. However, these methods only exploit a single modality. Recently, efforts have been made to integrate visual and tactile perception into imitation learning frameworks. Yet, they exhibit limitations in experimental settings: some are restricted to gripper tasks [27]–[29], some rely on bulky optical tactile sensors [5], [7], and others are confined to fingertips-only tactile sensing [30]. To that point, our work offers dual operational modes, employing distributed tactile sensors that provide full-palm sensing for the visuotactile mode, enabling challenging dexterous manipulation tasks.

III. Method

Dexterous in-hand manipulation requires rich hand-object interactions, where vision and touch complement each other. We propose Flow Before Imitation (FBI), a visuotactile imitation learning algorithm for this task. FBI takes two consecutive robot's state s_{t-1}, $s_t \in \mathbb{R}^{N_s}$, partially-observed point cloud frames P_{t-1}, $P_t \in \mathbb{R}^{N_p \times 3}$ and tactile readings $R_t \in \mathbb{R}^{N_r}$ as input, where N_s is the number of joints, N_p is the number of points in the downsampled point clouds (Section III-A), N_r is the number of contact keypoints (Section III-C). Such multimodal data are first processed by the multimodal encoders (Section III-A). After that, the encoded features are fused and passed as conditions to a shortcut model [13] to predict the action series $\mathbf{A}_t \in \mathbb{R}^{H \times d_a}$ (Section III-B), where H is the horizon of the action series, d_a is the dimension of each action.

While visuotactile systems excel in ideal conditions, we often encounter hardware limitations in real-world deployments, as distributed tactile sensors may be unavailable or prohibitively expensive. To enhance accessibility and lower the barriers to entry for labs and industries without tactile hardware, our method can maintain the capabilities using only visual input. Thanks to the dynamic perspective, tactile readings R_t can be predicted from the point cloud flow $f_{t-1 \to t} \in \mathbb{R}^{N_p \times 3}$ by the Flow2Tactile Module (Section III-C). An overview of our method is shown in Figure 2.

A. Multimodal Encoders

1) Robot State Encoder: We use a 3-layer MLP as state encoder $\boldsymbol{f}_s(\cdot)$ to encode the robot's two consecutive proprioception s_{t-1}, s_t into a compact state feature $\mathcal{F}_s \in \mathbb{R}^{d_s}$.

2) Visual Encoder: Given the two-frame point clouds of the hand and the object, we first crop and downsample them into N_p points and forward them to a lightweight MLP encoder $\boldsymbol{f}_v(\cdot)$. The lightweight encoder, consisting of a three-layer MLP, a max-pooling function, and an output MLP layer for feature dimension reduction, takes the downsampled point cloud P_{t-1} and P_t as input and outputs the compact feature $\mathcal{F}_v \in \mathbb{R}^{d_v}$, where d_v is the dimension of the visual feature. The visual feature will be fused with the tactile feature to form the visuotactile feature.

3) Tactile Encoder: To process the tactile readings, we use a four-layer MLP as the tactile encoder $\boldsymbol{f}_t(\cdot)$ to encode tactile feature $\mathcal{F}_{tac} \in \mathbb{R}^{d_{tac}}$, where d_{tac} is the dimension of the tactile feature. In the vision-only setting, the tactile readings come from the prediction results of the Flow2Tactile

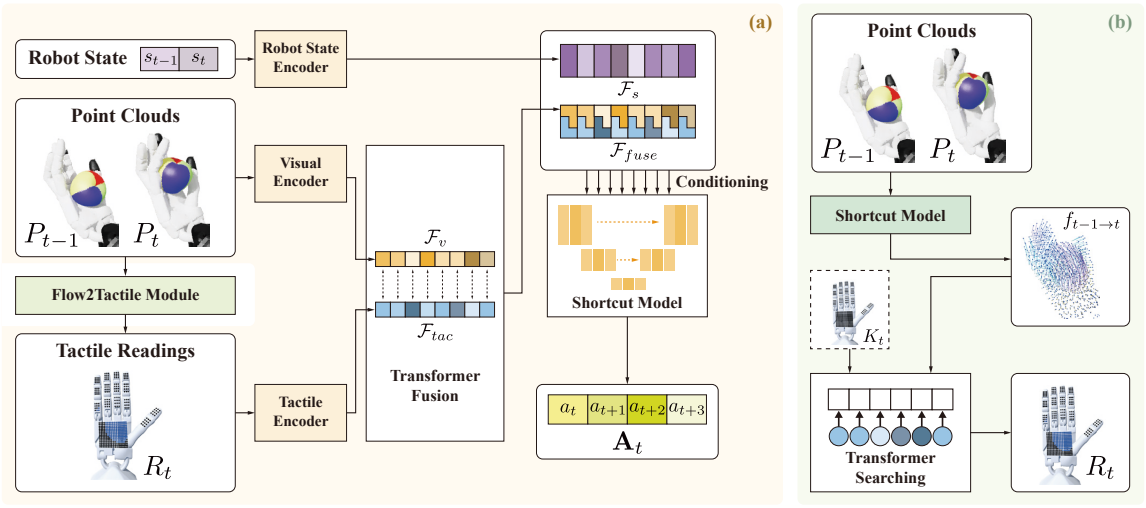


Fig. 2: **Overview of FBI pipeline.** (a) The overview of our pipeline, including multimodal encoders, a visuotactile feature fusing module, and the policy generation module. (b) The Flow2Tactile module that predicts the tactile readings, making it compatible with vision-only scenarios.

module. In the visuotactile setting, a physical tactile sensor is available, which provides direct readings from the sensor.

B. Visuotactile Policy Generation

With the visual feature $\mathcal{F}_v$ and tactile feature $\mathcal{F}_{tac}$, we can obtain the visuotactile features and use them as conditions for the policy generation model. In this work, we adopt a shortcut model [13] to predict action series $\mathbf{A}_t$. The shortcut model is a flow-matching-based generative model that enables high-quality one-step sampling. In the training phase, starting from an initial inference step $d\tau$, the model optimizes both flow matching loss and self-consistency loss simultaneously to improve its performance and increase the effective inference step size, ultimately achieving single-step generation ($d\tau = 1$). In the sampling phase, given condition C and a desired inference step size $d\tau$, the traditional flow-matching model can generate target distributions $p_\theta(x|C)$ from a standard normal distribution $x \sim \mathcal{N}(0, I)$. Starting from a Gaussian noise x^0, it utilizes a U-net [31] as the velocity predicting network $v_\theta(\cdot)$ to recursively predicts the velocity at denoising time step $\tau \in [0, 1)$ until finally reach the clear action x^1 at denoising time step 1:

$$x^{\tau+d\tau} = x^\tau + v_\theta(x^\tau, \tau, d\tau, C)d\tau. \tag{1}$$

In the shortcut model, which is one-step generation, we have $\tau = 0, d\tau = 1$.

1) Fusing Tactile Features with Vision Features: The encoded visual feature $\mathcal{F}_v$ and tactile feature $\mathcal{F}_{tac}$ are forwarded to a transformer fusion module $T_f(\cdot)$ which will take $\mathcal{F}_v$ as the template feature, $\mathcal{F}_{tac}$ as the searching feature and output the visuotactile feature. Considering visual observation is also crucial for environmental perception, we concatenate the visuotactile feature with the visual feature to form the final fused feature $\mathcal{F}_{fuse}$.

2) Predicting Actions from Multimodal Conditions: In our case, the model condition C consists of $\mathcal{F}_{fuse}$ and the robot state features $\mathcal{F}_s$. The velocity output by $v_\theta(\cdot)$ represents the denoising direction from the noisy action series $\mathbf{A}^0$ to the predicted action series $\mathbf{A}_t$. With the implementation of the shortcut model, we can use the $v_\theta(\cdot)$ **only once** to complete the prediction:

$$\mathbf{A}_t = \mathbf{A}^\tau + v_\theta(\mathbf{A}^\tau, \tau, d\tau, \mathcal{F}_s, \mathcal{F}_{fuse})d\tau, \tag{2}$$

where $\tau = 0, d\tau = 1$ in our case. Here, t denotes the time frame in the task progress and τ denotes the denoising time step. The benefits of the feature fusion on our policy will be included in Section IV-E.

3) Training Objective: We follow the main idea of training shortcut models to train our shortcut policy for one-step action generation. We first determine a minimum inference step size $d\tau$ and optimize both **flow matching loss** and **self-consistency loss** simultaneously. Assume that $\tau \in [0, 1)$ is the randomly selected variable, $\mathbf{A}^0 \sim \mathcal{N}(0, I)$ is the starting action Gaussian noise, $\mathbf{A}^1$ is the ground-truth action series, $\mathcal{F} = \{\mathcal{F}_s, \mathcal{F}_{fuse}\}$ is the conditional feature during the training phase, $v_\theta(\cdot)$ is the velocity predicting model. The loss function should be given as:

$$\mathcal{L} = \mathcal{L}_{FM} + \mathcal{L}_{SC}, \tag{3}$$

$$\mathcal{L}_{FM} = MSE\left((\mathbf{A}^1 - \mathbf{A}^0), v_\theta(\mathbf{A}^\tau, \tau, d\tau, \mathcal{F})\right), \tag{4}$$

$$\mathcal{L}_{SC} = MSE\left(v_\theta(\mathbf{A}^\tau, \tau, 2d\tau, \mathcal{F}), v_{target}\right), \tag{5}$$

$$v_{target} = \left[v_\theta(\mathbf{A}^\tau, \tau, d\tau, \mathcal{F}) + v_\theta(\mathbf{A}^{\tau+d\tau}, \tau+d\tau, d\tau, \mathcal{F})\right]/2, \tag{6}$$

$$\mathbf{A}^{\tau+d\tau} = \mathbf{A}^\tau + v_\theta(\mathbf{A}^\tau, \tau, d\tau, \mathcal{F})d\tau. \tag{7}$$

C. Flow2Tactile Module

Dense contact information is crucial for contact-rich manipulation tasks. However, acquiring it requires high-precision tactile sensors, which are often unavailable in the real world. Therefore, we leverage a **Flow2Tactile module** to predict dense contact states using object state flows. In doing so, our method can also work in the vision-only mode.

Specifically, an object state flow $f_{t-1\to t}$ is first predicted by a flow prediction model using two frames of point clouds P_{t-1} and P_t to represent how P_{t-1} transforms into P_t. Here, we implement the flow prediction model as a shortcut model due to its high capacity and fast inference. Then, a pre-trained transformer searching model $T_s(\cdot)$ predicts

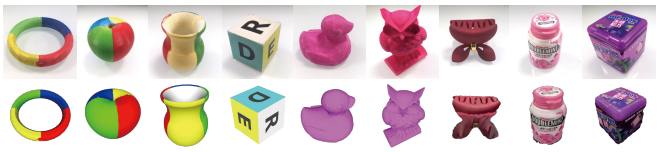


Fig. 3: Object dataset appearance in the real world (top row) and simulation (bottom row).

the tactile readings R_t on a pre-defined layout of contact keypoints utilizing $f_{t-1\to t}$ and the current coordinate of contact keypoints $K_t \in \mathbb{R}^{N_k\times 3}$, obtaining the dense contact state at time frame t. Here, N_k is the number of contact keypoints. The formation of the tactile processing module should be given as:

$$R_t = T_s(K_t, f_{t-1\to t}). \tag{8}$$

The contact keypoints are designed to comprehensively cover the hand's palm side, considering the physical parameters of real-world tactile sensors. This allows one to fit their real sensor deployments into our layout by finding a suitable correspondence between the real sensors and our predefined contact keypoint layout. Take the Shadow Hand [32] as an example, a set of 456 points, consisting of 12 keypoints on each finger link and 288 keypoints on the palm, is used as our contact keypoints (Figure 7). We use forward kinematics to calculate the coordinates $K_t \in \mathbb{R}^{456\times 3}$ of the contact keypoints at time frame t. For different robot hands, we need to customize a keypoint layout to cover the contact areas specific to each hand.

1) Training Objectives: The pre-trained Flow2Tactile module consists of two parts: flow prediction and a transformer searching module. For flow prediction, we make adjustments to Eq.3 by implementing the flow matching loss as the Chamfer Distance between point clouds:

$$\mathcal{L}_{FM} = CD\left(P_t, (P_{t-1} + f_{t-1\to t})\right). \tag{9}$$

The self-consistency loss is similar to Eq. 5, only to substitute $\mathbf{A}^\tau$ to f^τ, $\mathcal{F}$ to P, and the loss function for flow prediction is the sum of these two losses.

For the transformer searching module, we compute the loss between the output and the ground-truth tactile readings R_{gt}, which is calculated in simulation (detailed in Section IV-B). The loss function is given as:

$$\mathcal{L}_T = MSE\left(R_{gt}, T_s(K_t, f_{t-1\to t})\right). \tag{10}$$

IV. Experiments

In this section, we systematically evaluate our method in both simulation and real experiments. Specifically, we evaluate FBI in 2 in-hand manipulation tasks on various objects in both simulation and real-world settings. We also select 3 dexterous manipulation tasks from a public benchmark for fair comparison with baseline methods.

A. Task Setup

1) Object Dataset: An overview of our object dataset is shown in Figure 3. The object dataset contains objects of various shapes, weights, and colors. It has 6 3D-printed objects of both simple and complex shapes. The symmetric shapes are colored to indicate the orientation. We also have 3 daily objects to test the adaptability of our algorithm.

2) Robot Hand: For both simulation and real-world experiments, we utilized the Shadow Hand [32], a five-fingered robotic hand with 24 degrees of freedom.

3) Task Description: We evaluate FBI on two in-hand tasks: **Push** (moving objects to target positions) and **Reorientation** (adjusting orientations to target pose) in both simulation and real-world. Three Adroit tasks (**Door**, **Hammer**, **Pen**) [14] using Shadow Hand in MuJoCo [33] are also selected, following prior benchmarks [15], [16].

4) Evaluation Metric and Goal Definition: Success Rate (SR) is used for all tasks. Push success: object-target position $\leq$ 1mm; Reorientation success: orientation difference $\leq$ 0.1rad; Adroit tasks retain the original criteria. SR is auto-calculated in simulation and manually assessed in the real world. In our own tasks, the target goal is a synthetic object point cloud transformed into the desired pose, which is then concatenated with the observed scene point cloud and fed into the model. For Adroit tasks, goals (door, hammer, pen pose) are involved in the scene point cloud.

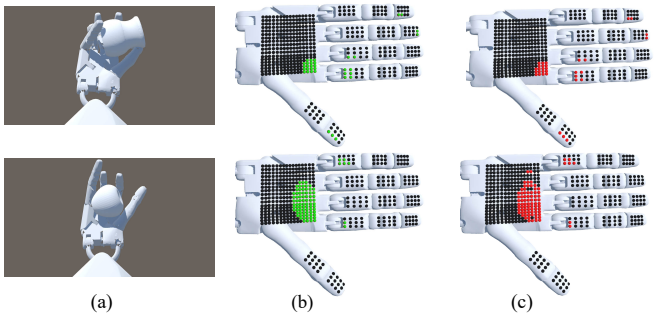
B. Training in Simulation

We simulate **In-hand Push** and **In-hand Reorientation** in Isaac Sim 4.1.0 [34] (25 Hz control, 120 Hz simulation). We first train state-based PPO agents [35] with a learning rate of 5e-4 and training epochs 1e4, and they achieve an average success rate of 65.7% for in-hand reorientation and 96.8% for in-hand push. Then, we use them to collect 5,000 successful trajectories, discarding failed ones as training demonstrations for each object. For **Adroit** tasks, in MuJoCo, a VRL3 [36] agent generates 10 successful trajectories per task. Finally, we train FBI for 300 epochs on the first two tasks and 3000 epochs on Adroit to ensure convergence.

To train the Flow2Tactile Module, we use ZeMa [37], a high-fidelity contact simulator widely used in real-world scenarios [12], [38], to compute tactile readings as training labels in simulation, as Isaac Sim lacks support for dense tactile sensing. We first record the hand and object states from RL-generated demonstrations, then replay the trajectories in ZeMa to compute frame-wise contact forces via Incremental Potential Contact [39], and finally binarize them to obtain the ground-truth tactile readings. Using 1,000 trajectories per object (average object density $463.1 kg/m^3$) for training, the Flow2Tactile module achieves 85.5% prediction accuracy on unseen data. Qualitative results of ZeMa-generated and predicted tactile readings are shown in Figure 4.

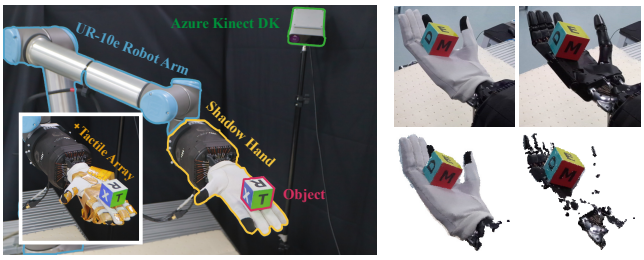
C. Real-World Deployment

1) Hardware Setup: A Kinect Azure camera is used to capture visual observations. To mitigate metallic reflections degrading point clouds, we equip the Shadow Hand with a glove (Figure 5). The glove is tightly fitted, with a thickness of less than 2mm, resulting in minimal deformation. For the visuotactile mode, RunesKee tactile sensors (0∼5N range, 0.1N resolution, 20 Hz communication frequency) are bought from *Taobao*, molded on a flexible printed circuit (FPC), and attached to finger links and the palm. The layout of the sensor is customized to fit the robot hand, which is shown in Figure 6(a). It features 8 sensors on the first phalanx of



(a) (b) (c)

Fig. 4: Qualitative results of Flow2Tactile Module. (a): The rendering result. (b): Ground-truth tactile readings (ZeMa). (c): Predicted tactile readings (Flow2Tactile Module).



(a) (b)

Fig. 5: (a): Real-world setup for vision-only mode and visuotactile mode. (b): Images and point clouds obtained by Kinect Azure before (right) and after (left) wearing gloves. Images are cropped versions that serve as input for image-based baselines.

each finger, 3 on the second phalanx except for the thumb, and 6 on the third phalanx. In total, there are 82 sensors on the fingers and 66 sensors on the palm, totaling 148 sensors. To fit our real sensor into the contact keypoints, we establish a mapping from the tactile sensors to the contact keypoints. Specifically, as shown in Figure 6(b), we divide the hand into several square regions, and calculate the contact keypoints readings using tri-linear interpolation based on the positions and actual readings from the real-world tactile sensors.

2) Sim-to-Real Transfer: FBI can be directly employed in real-world experiments. On the visual side, thanks to the glove coverage, the real-world point cloud quality is sufficient for direct transfer (Figure 5). On the tactile side, with a correspondence between the real sensors and contact keypoints, we can easily convert sensor readings into contact keypoint readings, thanks to the binary tactile reading design. On the action side, to make the robot's behavior smoother, we apply an exponential moving average to the predicted actions and an action interpolation between the current and target joint poses. After other sim-to-real alignments including point cloud registration, hand-eye calibration, and system identification, FBI can successfully perform dexterous manipulation tasks in the real world, with target joint states generated at approximately 20 Hz, accounting for both point cloud transmission time and model inference time.

D. Experiment Results

1) Baselines: We compare against four baselines: DP3 (point cloud diffusion) [15], ManiCM (point cloud consistency model) [16], Ada-Flow (image-based flow matching) [17], and Consistency Policy (image-based consistency diffusion) [18], selected to emphasize visuotactile feature fusion benefits. To further inspect the benefits of incorporating tactile information, we also evaluate our framework using only visual observations and robot states, and report the results in table V. For image-based methods, inputs are cropped to hand-object regions with uniform resolution. It is worth noting that, while some visuotactile-based imitation learning methods have emerged recently, they often focus on simpler gripper tasks, rely on bulky optical tactile sensors, or use fingertip-only tactile sensing (Section II). Meanwhile, our tasks primarily focus on in-hand manipulation, which includes rich interactions between the palm and objects. Adapting these methods to our task settings would be inequitable, so we do not include them for comparison.

(a) (b)

Fig. 6: (a): Tactile sensor layout on the Shadow Hand: palm side (left, sensor-covered) and dorsal side (right, no sensors). (b): Regions for tactile readings mapping. Same-color blocks mark corresponding regions.

2) Simulation Results: For simulation tasks, three seeds (0,1,2) are tested for 20 episodes every 20 training epochs, with top-5 success rates averaged per seed. We report the mean and standard deviation of success rates across three seeds. Results in Table I reveal that, when averaged over all tasks, FBI achieves improvements of 16.6% (Vision-Only) to 18.4% (Visuotactile) compared to DP3, with minor variance. For In-Hand Reorientation, FBI surpasses DP3 with a 19.3% (Vision-Only) to 21.4% (Visuotactile) improvement averaged over 9 objects, indicating that after fusing dense contact states with visual information, our algorithm demonstrates superior performance, particularly in handling more complex dexterous tasks. With tactile information and higher inference speed, FBI can lower the chance of objects getting stuck during the process, reach the target more smoothly, and reduce the occurrence of objects slipping from the hand.

3) Real-World Experiment Results: We evaluate each real-world task with 20 trials and count the success rates. We maintain the same hardware settings for every task to ensure a fair comparison. Results in Table II indicate that our results in the simulation experiments remain similar in real-world settings. Moreover, with the implementation of tactile information and the shortcut model, the manipulation process in the real world becomes faster, smoother, and more reliable. Qualitative results shown in Figure 7 illustrate two successful manipulation processes implemented in the real world.

4) Performance Discrepancy Analysis: Comparing Table I and Table II, all baselines exhibit a performance degradation in real-world success rates, even after our careful sim-to-real

Algorithm \ Task	NFE	In-hand Reorientation									In-hand Push	Adroit			Average
		Cube	Apple	Vase	Ring	Duck	Owl	daily object1	daily object2	daily object3	Balls	Hammer	Door	Pen	
FBI (Vision-Only)	1	80±3	86±2	74±5	**76±1**	**28±5**	27±4	65±4	39±2	**25±7**	95±3	**100±0**	**77±3**	69±4	64.7
FBI (Visuotactile)	1	**84±4**	**90±2**	**80±2**	**76±8**	**28±2**	**29±3**	**68±2**	**42±5**	22±10	**97±1**	**100±0**	75±2	**73±3**	**66.5**
DP3	10	51±9	53±12	44±2	50±5	20±5	18±3	49±2	24±5	18±6	84±7	**100±0**	69±5	45±5	48.1
ManiCM	1	50±3	50±10	46±3	53±2	17±2	15±6	47±6	25±7	16±4	86±5	**100±0**	68±3	43±6	47.4
Ada-Flow	1.82	15±12	30±2	21±4	15±7	1±1	3±2	30±7	25±2	10±3	44±15	50±12	52±3	28±4	24.9
Consistency Policy	1	19±8	32±3	20±3	18±6	1±2	2±0	25±10	26±3	12±3	48±10	55±6	50±7	26±5	25.7

TABLE I: Simulation Results. All baselines are run under the same simulation parameters to ensure a fair comparison. Lighter green indicates the second-best.

Algorithm \ Task	NFE	In-hand Reorientation									In-hand Push	Average
		Cube	Apple	Vase	Ring	Duck	Owl	daily object1	daily object2	daily object3	Balls	
FBI (Vision-Only)	1	**45.0**	**50.0**	35.0	**40.0**	**20.0**	15.0	**30.0**	**25.0**	15.0	60.0	33.5
FBI (Visuotactile)	1	**45.0**	**50.0**	**40.0**	**40.0**	15.0	**20.0**	**30.0**	**25.0**	**20.0**	**65.0**	**35.0**
DP3	10	10.0	15.0	5.0	5.0	0.0	0.0	10.0	5.0	0.0	20.0	7.0
ManiCM	1	25.0	25.0	30.0	15.0	10.0	5.0	20.0	10.0	15.0	30.0	18.5
Ada-Flow	1.32	10.0	10.0	10.0	15.0	0.0	0.0	15.0	15.0	5.0	25.0	10.5
Consistency Policy	1	15.0	10.0	15.0	15.0	5.0	0.0	10.0	15.0	10.0	20.0	11.5

TABLE II: Real-World Results. FBI outperforms all baselines in all tasks, achieving 15.0% (Vision-Only) to 16.5% (Visuotactile) improvement compared to the previous SOTA baseline, ManiCM. Lighter green indicates the second-best.

transfer (Section IV-C.2). It is mainly due to the challenging nature of our benchmark tasks, which require multiple and continuous contact. Even the most advanced contact modeling technique, IPC [39], continues to struggle with such scenarios. Unlike dexterous grasping or other quasi-static manipulation tasks, the simulation errors of long-horizon in-hand manipulation accumulate along with motion evolution. Although imperfect, the sim-to-real paradigm remains the most practical way to obtain training data for learning a policy for in-hand manipulation. Therefore, it is urgent to find more effective ways to enhance data collection, which is worth attracting more attention from the research community.

E. Ablation Study

In the ablation study, we analyze the impact of data scale, the importance of the Flow2Tactile module, tactile input variations, fusion method variations, generalization ability, and shortcut model inference speed effects.

1) Data Scale vs. Success Rate: To validate our data scale for contact-rich tasks, we analyze success rate vs. trajectory count using in-hand reorientation as a case study (Table III). Considering the task complexity, it does not show promising results after 100 data samples. The influence of the data sample number converges after 5000 trajectories.

	Number of Successful Trajectories								
	10	100	500	1000	3000	4000	5000	6000	8000
SR	0.0	4.3	16.0	27.6	45.1	51.8	55.6	56.2	56.6

TABLE III: Relationship between Success Rate (SR) and the number of successful trajectories. We test FBI (Vision-Only) on the In-hand Reorientation task in simulation. Success Rate is averaged across 9 objects.

2) Importance of Flow Prediction in Tactile Information Generation: We compare our method to other contact state generation methods to evaluate the Flow2Tactile module: 1) Flow2Tactile (Ours): the contact states are generated by searching the pre-placed keypoints with the predicted flow; 2) Point Cloud (PC) to Tactile: the contact states are generated by searching the pre-placed keypoints with two-frame point clouds directly; 3) TOCH [40]: a hand-object contact modeling method based on spatio-temporal correspondences, yet suffers from heavy optimization and slow real-time inference 4) Ground Truth: We use the ground-truth tactile readings R_{gt} (mentioned in Section III-C) calculated in simulation. Results are shown in Table IV.

Algorithm \ Task	In-hand Reorientation	In-hand Push	Adroit	Average
Flow2Tactile (Ours)	**55.6**	**95.0**	82.0	**64.7**
PC to Tactile	47.6	90.0	77.0	57.6
TOCH	53.9	94.0	**83.3**	63.8
Ground Truth	58.3	97.0	84.0	67.2

TABLE IV: Success Rates on different tactile generation methods. We include the ground-truth tactile readings R_{gt} as a reference.

Algorithm \ Task	In-hand Reorientation	In-hand Push	Adroit	Average
Dense Binary (Ours)	55.6	**95.0**	82.0	64.7
Dense Continuous Contact	**56.7**	93.0	**83.3**	**65.6**
Sparse Contact	38.2	87.0	73.3	50.1
W/o Contact	36.4	87.0	70.3	48.1

TABLE V: Success Rates on different tactile information. Lighter green indicates the second-best.

3) Different Forms of Tactile Information: To demonstrate the benefits of the **dense contact** information, we conduct experiments on different forms of tactile input: 1) Dense Binary Contact (Ours): The contact states are binary tactile readings on 456 contact keypoints; 2) Dense Continuous Contact: The contact states are continuous force readings applied to tactile sensors; 3) Sparse Binary Contact: The contact states are link-wise binary tactile readings between objects and the 24 Shadow Hand links; 4) W/o Contact: The policy model works without contact conditions, only with **visual cues** and **prioperception**. Results in Table V indicate that dense tactile information, either binary or continuous, greatly enhances the model's performance in dexterous tasks. In contrast, joint-wise sparse tactile information only offers slight gains. Although dense continuous contact provides the best performance, we choose dense binary contact as our representation because it simplifies sim-to-real transfer and improves the accuracy of model predictions, as binary predictions are easier for neural networks than regression.

4) Benefits of the Transformer-based visuotactile fusing approach: To demonstrate the effectiveness of the visuo-

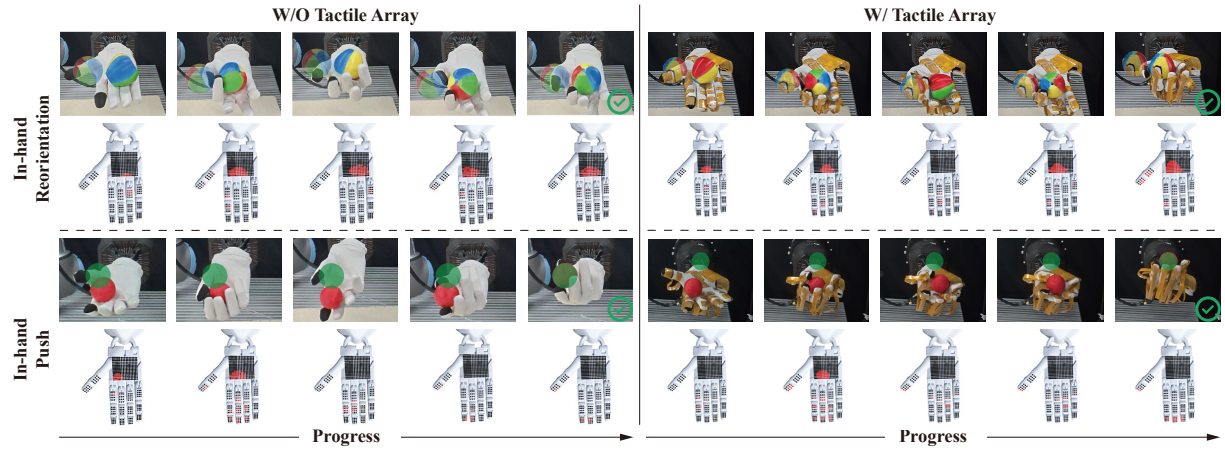


Fig. 7: **Qualitative results of the In-hand Reorientation task and the In-hand Push task.** We demonstrate both the vision-only mode (left) and the visuotactile mode (right). We capture and display five key frames from each manipulation process, along with the corresponding tactile readings.

tactile fusing module, we conduct experiments on different fusing methods: 1) Transformer Fusion (Ours): The visual and tactile features are forwarded to our transformer fusion module; 2) MLP Fusion: We use a multi-layer MLP to fuse the features; 3) Add: We add the visual and tactile features to get the visuotactile feature. Results are shown in Table VI.

Algorithm \ Task	In-hand Reorientation	In-hand Push	Adroit	**Average**
Transformer Fusion (Ours)	**55.6**	**95.0**	**82.0**	**64.7**
MLP Fusion	48.1	92.0	77.3	58.2
Add	43.2	89.0	75.7	54.2

TABLE VI: Success Rates on different fusing method.

5) Generalization to Unseen Objects: FBI demonstrates promising generalization to unseen object sizes/shapes. In cube reorientation, both FBI and DP3 succeed on original objects, but FBI outperforms DP3 on novel variants in zero-shot tests (Table VII), attributed to the contact modeling. We use FBI (Vision-Only) for fair comparison.

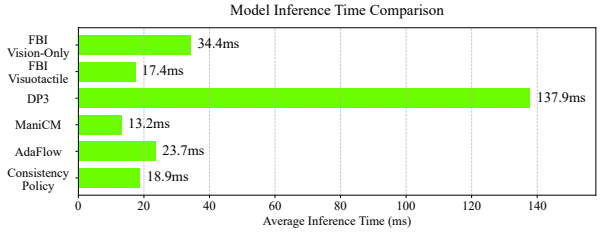
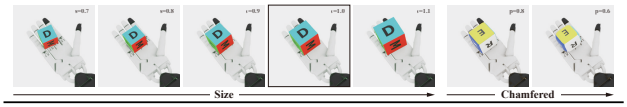


Fig. 8: Average real-world inference latency across all baselines. We measure from input reception (images/point clouds/states) to action series output for fairness.

6) Details on The Model's Inference Speed: We detail the inference speed in the real world and justify the use of shortcut models [13]. We report the **average model inference time** (denoted as AVG Time) for all baselines on an NVIDIA RTX 4090 GPU in real-world experiments in Figure 8. FBI (Vision-Only) achieves 20 Hz control (51.2ms per action series), including environment interaction (16.8ms) and model inference (34.4ms). it spends 17.4ms on the Flow2Tactile Module (flow prediction: 11.2ms, tactile reading prediction: 6.2ms) and 15.0ms on the visuotactile policy (Transformer fusion: 4.8ms, shortcut policy: 10.2ms), with 2.0ms overhead. FBI (Visuotactile) reduces the model inference time to 17.4ms (15.2ms for the policy and 2.2ms overhead) by skipping the Flow2Tactile Module. In contrast, DP3 [15] requires 137.9ms for model inference, and even its simplified variant (Simple DP3) spends 83.6ms. Replacing FBI's shortcut models with 10-step DDIM [41] would inflate latency to 261.4ms, underscoring the necessity of our design.



Unseen Shape	Algorithms	NO.1	NO.2	NO.3	NO.4	NO.5
0.9× Size	**FBI (VO)**	✓	✓	✓	✓	✓
	DP3	✓	✓	✓	✗	✓
0.8× Size	**FBI (VO)**	✓	✓	✗	✓	✓
	DP3	✗	✓	✗	✗	✓
0.75× Size	**FBI (VO)**	✓	✗	✗	✗	✓
	DP3	✗	✗	✗	✗	✗
1.1× Size	**FBI (VO)**	✓	✓	✓	✗	✓
	DP3	✓	✓	✗	✗	✓
Chamfered Cube (Param.0.8)	**FBI (VO)**	✓	✗	✓	✗	✓
	DP3	✗	✗	✗	✗	✓
Chamfered Cube (Param.0.6)	**FBI (VO)**	✓	✗	✗	✗	✗
	DP3	✗	✗	✗	✗	✗

TABLE VII: Size and shape generalization on cubes. We test each unseen size/shape with 5 distinct initial and target poses, denoted as No. 1-No. 5. VO stands for Vision-Only.

V. Conclusion

This work introduces Flow Before Imitation (FBI), a visuotactile fusion framework that advances in-hand manipulation by dynamically linking tactile interactions to object motion dynamics. FBI's dynamics-aware latent model enables synergy between vision and touch, supporting robust performance with or without physical tactile sensors. Evaluations across 5 dexterous tasks in both simulated and real-world settings demonstrate FBI's superiority, achieving an 18.4% higher success rate than state-of-the-art baselines, with notable gains (21.4%) in in-hand reorientation. Crit-

ically, FBI maintains stability under partial sensor failure, enhancing deployability in practical settings.

ACKNOWLEDGMENT

This work was supported by the National Key Research and Development Project of China (Grant No. 2024YFB4707603), Science and Technology Major Project of Jiangsu Province (No. BG2024041), the Shanghai Commitee of Science and Technology, China (Grant No. 24511103200) Shanghai Artificial Intelligence Laboratory, XPLORER PRIZE grants, the Shanghai Municipal Education Commission (No. 2024AIYB010), the Fundamental Research Funds for the Central Universities (YG2025LC03).

REFERENCES

[1] K. Xu, Z. Hu, R. Doshi, A. Rovinsky, V. Kumar, A. Gupta, and S. Levine, "Dexterous manipulation from images: Autonomous real-world rl via substep guidance," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5938–5945.

[2] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang, "Dexmv: Imitation learning for dexterous manipulation from human videos," *arXiv preprint arXiv:2108.05877*, 2021.

[3] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[4] Z.-H. Yin, B. Huang, Y. Qin, Q. Chen, and X. Wang, "Rotating without seeing: Towards in-hand dexterity through touch," *arXiv preprint arXiv:2303.10880*, 2023.

[5] M. Yang, C. Lu, A. Church, Y. Lin, C. Ford, H. Li, E. Psomopoulou, D. A. Barton, and N. F. Lepora, "Anyrotate: Gravity-invariant in-hand object rotation with sim-to-real touch," *arXiv preprint arXiv:2405.07391*, 2024.

[6] K.-W. Lee, Y. Qin, X. Wang, and S.-C. Lim, "Dextouch: Learning to seek and manipulate objects with tactile dexterity," *IEEE Robotics and Automation Letters*, 2024.

[7] H. Qi, B. Yi, S. Suresh, M. Lambeta, Y. Ma, R. Calandra, and J. Malik, "General in-hand object rotation with vision and touch," in *Conference on Robot Learning*. PMLR, 2023, pp. 2549–2564.

[8] I. Guzey, B. Evans, S. Chintala, and L. Pinto, "Dexterity from touch: Self-supervised pre-training of tactile representations with robotic play," *arXiv preprint arXiv:2303.12076*, 2023.

[9] I. Guzey, Y. Dai, B. Evans, S. Chintala, and L. Pinto, "See to touch: Learning tactile dexterity through visual incentives," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 13 825–13 832.

[10] Y. Yuan, H. Che, Y. Qin, B. Huang, Z.-H. Yin, K.-W. Lee, Y. Wu, S.-C. Lim, and X. Wang, "Robot synesthesia: In-hand manipulation with visuotactile sensing," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 6558–6565.

[11] S. Sundaram, P. Kellnhofer, Y. Li, J.-Y. Zhu, A. Torralba, and W. Matusik, "Learning the signatures of the human grasp using a scalable tactile glove," *Nature*, vol. 569, no. 7758, pp. 698–702, 2019.

[12] C. Jiang, W. Xu, Y. Li, Z. Yu, L. Wang, X. Hu, Z. Xie, Q. Liu, B. Yang, X. Wang *et al.*, "Capturing forceful interaction with deformable objects using a deep learning-powered stretchable tactile array," *Nature Communications*, vol. 15, no. 1, p. 9513, 2024.

[13] K. Frans, D. Hafner, S. Levine, and P. Abbeel, "One step diffusion via shortcut models," *arXiv preprint arXiv:2410.12557*, 2024.

[14] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," *arXiv preprint arXiv:1709.10087*, 2017.

[15] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy," *arXiv e-prints*, pp. arXiv–2403, 2024.

[16] G. Lu, Z. Gao, T. Chen, W. Dai, Z. Wang, W. Ding, and Y. Tang, "Manicm: Real-time 3d diffusion policy via consistency model for robotic manipulation," *arXiv preprint arXiv:2406.01586*, 2024.

[17] X. Hu, Q. Liu, X. Liu, and B. Liu, "Adaflow: Imitation learning with variance-adaptive flow-based policies," *Advances in Neural Information Processing Systems*, vol. 37, pp. 138 836–138 858, 2024.

[18] A. Prasad, K. Lin, J. Wu, L. Zhou, and J. Bohg, "Consistency policy: Accelerated visuomotor policies via consistency distillation," *arXiv preprint arXiv:2405.07503*, 2024.

[19] S. P. Arunachalam, S. Silwal, B. Evans, and L. Pinto, "Dexterous imitation made easy: A learning-based framework for efficient dexterous manipulation," in *2023 ieee international conference on robotics and automation (icra)*. IEEE, 2023, pp. 5954–5961.

[20] S. Cruciani, C. Smith, D. Kragic, and K. Hang, "Dexterous manipulation graphs," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2040–2047.

[21] F. Khadivar and A. Billard, "Adaptive fingers coordination for robust grasp and in-hand manipulation under disturbances and unknown dynamics," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3350–3367, 2023.

[22] A. Hammoud, V. Belcamino, A. Carfi, V. Perdereau, and F. Mastrogiovanni, "In-hand manipulation planning using human motion dictionary," in *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2022, pp. 927–933.

[23] D. Antotsiou, G. Garcia-Hernando, and T.-K. Kim, "Task-oriented hand motion retargeting for dexterous manipulation imitation," in *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018, pp. 0–0.

[24] 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.

[25] Y. Su, X. Zhan, H. Fang, Y.-L. Li, C. Lu, and L. Yang, "Motion before action: Diffusing object motion as manipulation condition," *arXiv preprint arXiv:2411.09658*, 2024.

[26] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," *arXiv preprint arXiv:2210.02747*, 2022.

[27] H. Xue, J. Ren, W. Chen, G. Zhang, Y. Fang, G. Gu, H. Xu, and C. Lu, "Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation," *arXiv preprint arXiv:2503.02881*, 2025.

[28] B. Huang, Y. Wang, X. Yang, Y. Luo, and Y. Li, "3d-vitac: Learning fine-grained manipulation with visuo-tactile sensing," *arXiv preprint arXiv:2410.24091*, 2024.

[29] F. Liu, C. Li, Y. Qin, A. Shaw, J. Xu, P. Abbeel, and R. Chen, "Vitamin: Learning contact-rich tasks through robot-free visuo-tactile manipulation interface," *arXiv preprint arXiv:2504.06156*, 2025.

[30] T. Lin, Y. Zhang, Q. Li, H. Qi, B. Yi, S. Levine, and J. Malik, "Learning visuotactile skills with two multifingered hands," *arXiv preprint arXiv:2404.16823*, 2024.

[31] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *arXiv preprint arXiv:1505.04597*, 2015.

[32] ShadowRobot, "Shadow hand," https://www.shadowrobot.com/.

[33] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.

[34] NVIDIA, "Isaac sim - robotics simulation and synthetic data generation," https://developer.nvidia.com/isaac-sim, 2023.

[35] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[36] C. Wang, X. Luo, K. Ross, and D. Li, "Vrl3: A data-driven framework for visual deep reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32 974–32 988, 2022.

[37] W. Du, S. Yao, X. Wang, Y. Xu, W. Xu, and C. Lu, "Intersection-free robot manipulation with soft-rigid coupled incremental potential contact," *IEEE Robotics and Automation Letters*, 2024.

[38] Z. Yu, W. Xu, P. Xie, Y. Li, and C. Lu, "Dynamic reconstruction of hand-object interaction with distributed force-aware contact representation," *arXiv preprint arXiv:2411.09572*, 2024.

[39] M. Li, Z. Ferguson, T. Schneider, T. R. Langlois, D. Zorin, D. Panozzo, C. Jiang, and D. M. Kaufman, "Incremental potential contact: intersection-and inversion-free, large-deformation dynamics." *ACM Trans. Graph.*, vol. 39, no. 4, p. 49, 2020.

[40] K. Zhou, B. L. Bhatnagar, J. E. Lenssen, and G. Pons-Moll, "Toch: Spatio-temporal object-to-hand correspondence for motion refinement," *arXiv preprint arXiv:2205.07982*, 2022.

[41] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.