UMI-on-Air: Embodiment-Aware Guidance for Embodiment-Agnostic Visuomotor Policies

Harsh Gupta[†] Xiaofeng Gao[†] Huy Ha[‡] Chuer Pan[‡] Muqing Cao[†] Dongjae Lee[†]
Sebastian Sherer[†] Shuran Song[‡] Guanya Shi[†]

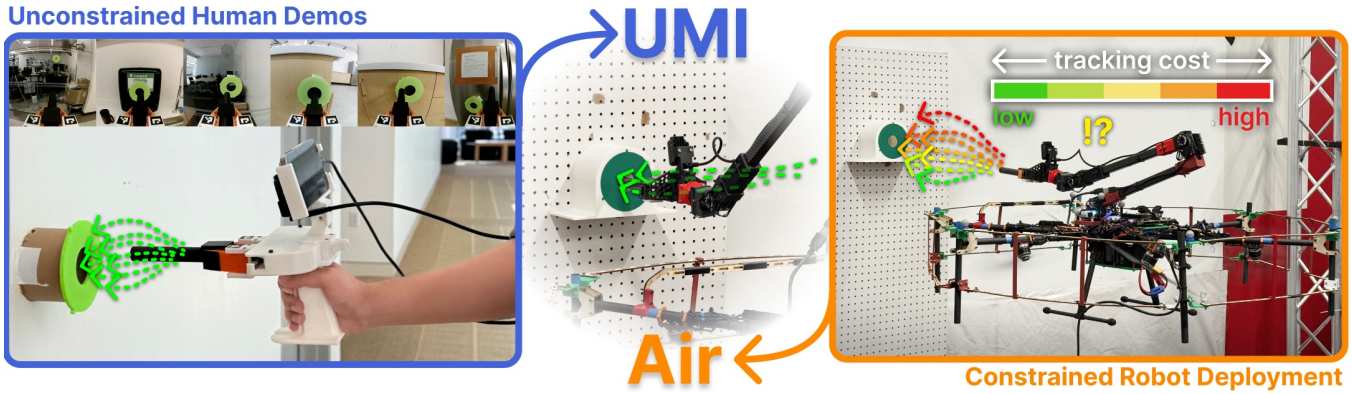

Fig. 1: **UMI-on-Air with Embodiment-Aware Guidance.** Standard UMI (Universal Manipulation Interface, [1], [2]) systems use **one-way** communication by sending high-level policy outputs to low-level controllers via end-effector trajectories—often suboptimal or even infeasible for a given embodiment. Our approach introduces **two-way** communication, letting the low-level controller steer UMI policies from actions with **high tracking cost** to those with **lower cost**, enabling more robust and high-performance cross-embodiment deployment.

***Abstract*— We introduce UMI-on-Air, a framework for embodiment-aware deployment of embodiment-agnostic manipulation policies. Our approach leverages diverse, unconstrained human demonstrations collected with a handheld gripper (UMI) to train generalizable visuomotor policies. A central challenge in transferring these policies to constrained robotic embodiments—such as aerial manipulators—is the mismatch in control and robot dynamics, which often leads to out-of-distribution behaviors and poor execution. To address this, we propose Embodiment-Aware Diffusion Policy (EADP), which couples a high-level UMI policy with a low-level embodiment-specific controller at inference time. By integrating gradient feedback from the controller's tracking cost into the diffusion sampling process, our method steers trajectory generation towards dynamically feasible modes tailored to the deployment embodiment. This enables plug-and-play, embodiment-aware trajectory *adaptation at test time*. We validate our approach on multiple long-horizon and high-precision aerial manipulation tasks, showing improved success rates, efficiency, and robustness under disturbances compared to unguided diffusion baselines. Finally, we demonstrate deployment in previously unseen environments, using UMI demonstrations collected in the wild, highlighting a practical pathway for scaling generalizable manipulation skills across diverse—and even highly constrained—embodiments. All code, data, and checkpoints will be publicly released after acceptance. Result videos can be found at umi-on-air.github.io.**

I. Introduction

There is a growing interest in extending manipulation beyond the lab and into more complex, dynamic settings. Among emerging embodiments, unmanned aerial manipulators (UAMs) hold particular promise. Essentially a manipulator with practically limitless reach, UAMs can access environments that are otherwise unreachable or unsafe—such as performing infrastructure maintenance atop towers or harvesting crops in cluttered orchards. Multiple research works have focused on those practical tasks, including non-destructive testing [3], painting [4], drilling [5], light bulb installation [6], etc. These applications highlight the potential of UAMs as an embodiment, but scalable visuomotor policy learning for UAMs remains limited.

A major bottleneck is data collection. Teleoperation is particularly challenging for UAMs due to expensive and fragile hardware and unintuitive interface. To address this, recent work explores cross-embodiment collection, with the Universal Manipulation Interface (UMI) [1] offering a portable, low-cost way to record demonstrations across environments. By decoupling demonstrations from specific robots, UMI enables training of embodiment-agnostic manipulation policies.

While UMI enables embodiment-agnostic manipulation policies, their success hinges on the embodiment's ability to execute the generated trajectories. Fixed-base arms with precise controllers are highly "UMI-able[1]", able to execute UMI policies as if they were the handheld gripper. In contrast, embodiments like UAMs face stringent physical and control constraints such as stability under aerodynamic disturbances [7], [8]. Without accounting for these constraints, UMI policies may yield trajectories that are infeasible, unsafe, or inefficient. Hence, the **central challenge** is then how to

[†]Carnegie Mellon University [‡]Stanford University

[1]We will formally define "UMI-able" in § IV, and Fig. 6 quantifies how "UMI-able" different embodiments are in simulation.

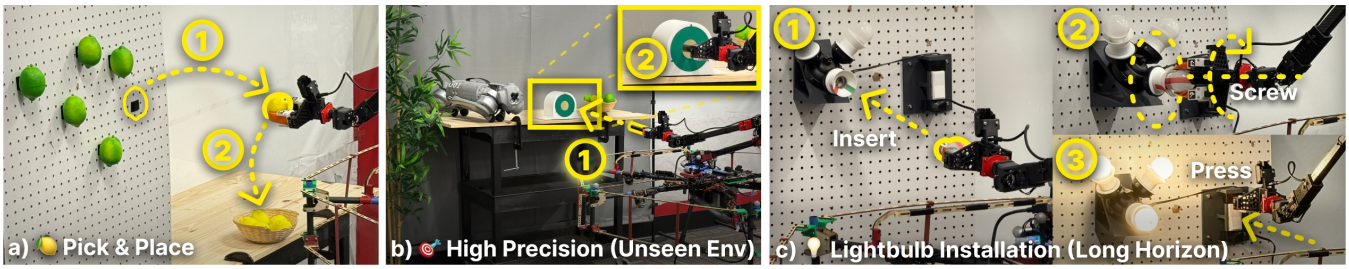

Fig. 2: **Aerial Manipulation Tasks.** Combining UMI and our embodiment-aware guidance approach enables scalable data-collection and robust deployment of fully-autonomous skills previously beyond reach. On our UAM, we showcase (a) lemon harvesting (must find ripe yellow ones), (b) high precision peg insertion in *unseen* environments, and (c) long-horizon light bulb installation tasks.

extend UMI beyond highly UMI-able robots to embodiments where control constraints fundamentally shape feasibility.

To address this challenge, we propose **Embodiment-Aware Diffusion Policy (EADP)**, where the key idea is to enable **two-way** communication between an embodiment-agnostic high-level manipulation policy and embodiment-specific low-level controllers (Fig. 1). Unlike standard UMI systems [1], [2] that rely on **one-way** communication by passing policy outputs directly to controllers, EADP lets low-level controllers actively guide the high-level policy through the denoising process, therefore producing end-effector (EE) trajectories that are more feasible for the target embodiment (e.g., a UAM).

Concretely, at each denoising step, the embodiment's controller evaluates the noisy EE trajectory with a tracking cost, measuring its feasibility under current constraints. By backpropagating this cost to the noisy action trajectory, the policy is guided toward action trajectories that are more feasible. By leveraging the multi-modality of UMI policies (from diverse human data and the diffusion architecture), EADP biases the action generation toward strategies best aligned with the embodiment's capabilities.

In summary, the work has three main contributions:

- We propose Embodiment-Aware Diffusion Policy, a framework that integrates embodiment-specific controller feedback into high-level trajectory generation by diffusion policies, enabling plug-and-play embodiment-aware trajectory guidance at test time.
- We introduce a simulation-based benchmark suite[2], which facilitates systematic investigation of the embodiment gap when using UMI demonstration data on embodiments with varying UMI-abilities.
- We present UMI-on-Air, a system that validates our method on challenging aerial manipulation tasks (Fig. 2), outperforming embodiment-agnostic baselines.

By closing the gap between embodiment-agnostic policies and embodiment-specific constraints, this work is a step towards making all robots more UMI-able, thus extending scalable, universal manipulation skills to robots and environments previously beyond reach.

II. Related Works

A. Mobile Manipulation

Ground-Based Manipulation. Ground-based mobile manipulation systems traditionally emerged to be designed to tailor to specific use cases [9], [10], [11], [12], [13]. This led to a strong reliance on task and motion planning and model-based control, to capture the unique embodiment kinematics and dynamics for the specific mobile system. Recent learning-based systems have shown success in leveraging behavior cloning [14], [15], [16], [17], [18], reinforcement learning (RL) [19], [20], [21], [22], combining reinforcement learning for locomotion and behavior cloning for manipulation [23], [2] as an alternative for ground-based mobile manipulation via imitation learning.

Aerial Manipulation. While aerial manipulation is a subset of mobile manipulation, it introduces distinct challenges compared to ground-based mobile embodiments, including disturbance near ground and wall, stability requirements, underactuated nonlinear dynamics, and strict payload constraints. Aerial manipulation has been demonstrated across a wide variety of applications, including inspection of surfaces [3], [24], writing and painting [25], [26], object grasping [27], [28], insertion [29], [30], and articulated object interaction [31], [32]. These successes have typically relied on specialized hardware systems coupled with carefully engineered control strategies for a specific task, but remain hard to scale to novel manipulation goals or environments. To support broader deployment across aerial tasks, recent research has shifted toward developing general-purpose frameworks that abstract away embodiment-specific dynamics—for example, EE-centric control interfaces [6], which decouples high-level decision making from low-level embodiment-specific actuation. Despite these advancements, robust and generalizable policies for such systems require training on large-scale *robot* data that span diverse objects, scenes, and flight conditions. However, data collection directly with UAMs is challenging due to the difficulty of deploying drones across diverse physical environments, motivating alternative strategies for data collection and deployment.

[2]All code, data, and checkpoints will be publicly released after acceptance. Result videos can be found at umi-on-air.github.io

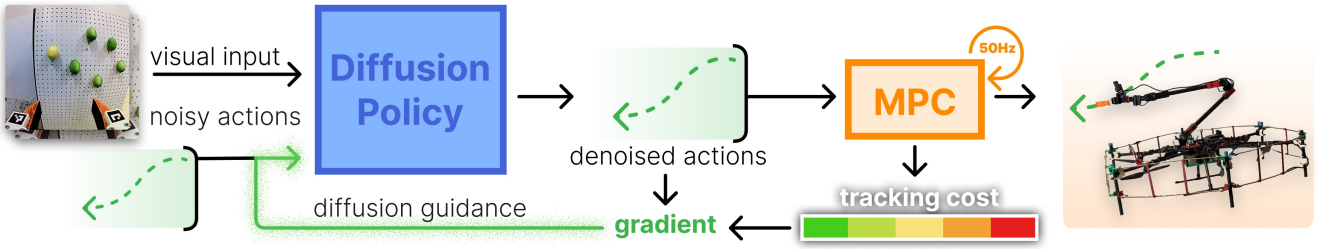

Fig. 3: **Embodiment-Aware Diffusion Policy.** Using UMI, we collect data for an embodiment-agnostic **Diffusion Policy**, which iteratively denoises actions from visual inputs. To produce more feasible actions, we add **gradients** of the MPC's tracking cost to the diffusion model's output at each iteration, steering the denoising process akin to classifier guidance. Finally, the guided action sequence is tracked by **MPC** at 50Hz.

B. Cross-embodiment Learning

Recent works have explored using large-scale cross-embodiment datasets as a pretraining strategy, involving data collected from various robotic embodiments and subsequently finetuned to accommodate specific hardware embodiments [33], [34], [35], [36]. These approaches rely on the assumption of a unified action space, which allows data sharing across robots with similar morphology, primarily robotic arms or mobile manipulators. Despite improvements in generalization, such methods require extensive embodiment-specific finetuning datasets to ensure policies adapt to the target embodiment.

An alternative strategy involves collecting demonstration data directly from the human-embodiment using intuitive handheld interfaces [1], [37], [38], [39], [40]. For instance, UMI [1] bridges the embodiment gap by minimizing differences between the observation space and action spaces of the hand-held gripper and the robot embodiment. This approach reduces the overhead associated with robotic data collection and enables large-scale, in-the-wild demonstrations without relying on physical robot hardware. While these methods have demonstrated impressive capabilities for tasks including precise or dynamic manipulation, policies trained directly from human demonstrations internalize action constraints reflective of human embodiment. As a result, these policies remain unaware of the distinct dynamics and physical limitations of embodiments like mobile manipulators [2], [6], where the EE cannot precisely track the generated action sequences, leading to unreliable execution.

To incorporate embodiment information into policy representations, recent works have developed specialized model architectures. Embodiment-aware policies leveraged graph neural networks (GNNs), explicitly modeling robot structures as graphs, with joints as nodes and links as edges [41], [42]. Subsequent works have explored transformer-based models, motivated by their superior representational capacity [43], [44], [45], [46]. Embodiment-aware architectures have demonstrated impressive zero-shot generalization capabilities within RL contexts by using large-scale training with extensive embodiment randomization, yet their adoption within imitation learning remains limited due to a lack of such data. In contrast, our work proposes incorporating embodiment-awareness during inference time by integrating feedback from a low-level embodiment-specific controller into a diffusion policy's trajectory generation process. By iteratively guiding diffusion-based sampling toward controller-feasible trajectories, we gain the benefits of abstraction from an ee-centric action space while producing trajectories that respect the robot's physical constraints.

III. Method

A. Data Collection Interface

We adopt the Universal Manipulation Interface (UMI) [1] paradigm for human demonstration collection: a lightweight, hand-held gripper with a wrist-mounted camera for egocentric observation, and a shared action interface expressed in the EE frame. This design enables in-the-wild data collection without requiring robot hardware, and aligns the training and deployment modalities by replicating the camera–gripper configuration on the robot.

In our configuration (Fig. 4), we make three key modifications to the original UMI for UAM deployment. First, we replace the GoPro with a lightweight OAK-1 W camera, which reduces payload while maintaining a wide field of view. Second, we downsized the finger geometry to reduce the inertia of the EE. Finally, we use an iPhone-based visual–inertial SLAM system to more accurately track the 6-DoF EE pose during data collection.

Each demonstration consists of synchronized egocentric RGB images, 6-DoF EE pose trajectories, and continuous gripper width tracked using fiducial markers on the fingers. These sequences form input–output pairs for policy learning: the input is an observation window consisting of images, relative EE poses, and gripper widths, while the output is a horizon of future actions given as relative EE trajectories and gripper widths. A conditional UNet-based [47] diffusion policy is trained on these pairs, enabling the generation of multimodal action sequences from the UMI demonstrations.

B. End-Effector-Centric Controllers

A key requirement for deploying embodiment-agnostic policies is a controller that interprets task-space reference trajectories $a = \{\boldsymbol{p}_t^r, \boldsymbol{R}_t^r\}_{t=1}^H$—a sequence of desired EE positions $\boldsymbol{p}^r \in \mathbb{R}^3$ and orientations $\boldsymbol{R}^r \in SO(3)$ over horizon H—into embodiment-specific actions. We adopt an *EE–centric* perspective: the high-level policy always produces EE reference trajectories, while the controller is responsible

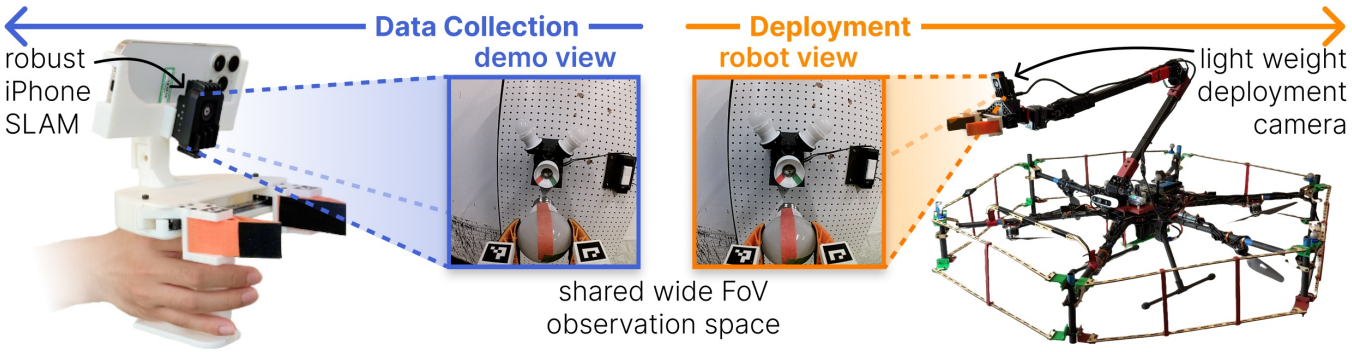

Fig. 4: **Data Collection to Deployment.** Our data collection setup contains an iPhone running SLAM tracking, a lightweight camera for deployment, and compliant, 3D-printed gripper fingers. By sharing the observation and action space between data collection and deployment time, we minimize the embodiment gap.

for realizing them subject to embodiment constraints. This abstraction supports a spectrum of controllers, ranging from simple inverse kinematics (IK) with velocity limits to full model predictive control (MPC).

To guide the diffusion policy toward embodiment-feasible behaviors, we define a tracking cost $L_{\text{track}}(a)$ that evaluates how well a given trajectory a can be executed by a particular controller. High tracking cost indicates segments of the trajectory that are hard to follow—due to dynamic infeasibility, underactuation, or control saturation—while low cost indicates better alignment with the embodiment's capabilities.

a) Inverse Kinematics with Velocity Limits: For tabletop manipulators and other robots with relatively simple dynamics, a lightweight controller can model the system well. At each step, the desired waypoint $(\boldsymbol{p}^r, \boldsymbol{R}^r)$ is mapped to a robot configuration $\boldsymbol{q} \in \mathbb{R}^n$—which may include both mobile base pose and arm joint angles—using the inverse kinematics function $\boldsymbol{f}_{\text{IK}}$, which maps a desired EE pose $(\boldsymbol{p}^r, \boldsymbol{R}^r)$ together with the current configuration $\boldsymbol{q}_t$ to a feasible robot configuration. We denote the per-step velocity bound as $\boldsymbol{\delta}_{\max} = \dot{\boldsymbol{q}}_{\max}\Delta t$, which accounts for the hardware velocity limit $\dot{\boldsymbol{q}}_{\max}$ and the controller timestep Δt. The forward kinematics $\boldsymbol{f}_{\text{FK}}(\boldsymbol{q})$ reconstructs the trajectory waypoint, and the tracking cost is the squared error between reconstructed and reference trajectories:

$$\boldsymbol{q}_{t+1} = \boldsymbol{q}_t + \text{clip}\left(\boldsymbol{f}_{\text{IK}}(a_t, \boldsymbol{q}_t) - \boldsymbol{q}_t,\ -\boldsymbol{\delta}_{\max},\ \boldsymbol{\delta}_{\max}\right) \tag{1}$$

$$L_{\text{track}}(a) = \sum_{t=1}^{H} \left\| \boldsymbol{f}_{\text{FK}}(\boldsymbol{q}_t) - a_t \right\|^2 \tag{2}$$

This provides a feasible sequence of configurations and a differentiable tracking cost L_{track}.

b) Model Predictive Controller: Richer controller instantiations can be used for robots that require accurate modeling of dynamics, such as UAMs. We adopt the EE–centric whole-body MPC from [6]. This controller coordinates UAV and manipulator motion by optimizing a finite-horizon cost function subject to dynamics and actuation constraints. The state and control variables are defined as:

$$\boldsymbol{x} := \begin{bmatrix} \boldsymbol{p} & \boldsymbol{R} & \boldsymbol{v} & \boldsymbol{\theta} \end{bmatrix}, \quad \boldsymbol{u} := \begin{bmatrix} \boldsymbol{\tau} & \boldsymbol{\theta}_{\text{cmd}} \end{bmatrix}, \tag{3}$$

where $\boldsymbol{v} \in \mathbb{R}^6$ is the body velocity (linear + angular), $\boldsymbol{\theta} \in \mathbb{R}^n$ are the manipulator's joint angles, $\boldsymbol{\tau} \in \mathbb{R}^6$ is the commanded wrench (forces and torques) and $\boldsymbol{\theta}_{\text{cmd}} \in \mathbb{R}^n$ are the commanded joint angles. Note that we adopt a similar UAM system with [6], which is a fully-actuated hexarotor, allowing us to send the commanded 6-dim control wrench directly.

The cost functions are defined in terms of errors between the predicted and reference values:

$$\boldsymbol{e}_p = \boldsymbol{p} - \boldsymbol{p}^r \tag{4a}$$

$$\boldsymbol{e}_R = \frac{1}{2}\left(\boldsymbol{R}^{r\top}\boldsymbol{R} - \boldsymbol{R}^\top\boldsymbol{R}^r\right)^\vee \tag{4b}$$

$$\boldsymbol{e}_v = \boldsymbol{v} - \boldsymbol{v}^r \tag{4c}$$

$$\boldsymbol{e}_\theta = \boldsymbol{\theta} - \boldsymbol{\theta}^r \tag{4d}$$

$$\boldsymbol{e}_u = \boldsymbol{u} - \boldsymbol{u}^r \tag{4e}$$

where $(\cdot)^r$ denotes reference values, and $(\cdot)^\vee$ denotes the vee-operator that maps a skew-symmetric matrix to $\mathbb{R}^3$. The default reference joint angles $\boldsymbol{\theta}^r$ are pre-defined, the reference velocity is set to $\boldsymbol{v}^r = [\boldsymbol{0}_6]$ and the reference control $\boldsymbol{u}^r = [\boldsymbol{0}_6, \hat{\boldsymbol{\theta}}]$ assumes zero wrench and current joint positions $\hat{\boldsymbol{\theta}} \in \mathbb{R}^n$.

The optimal control sequence is obtained by solving the following finite-horizon constrained optimization:

$$\boldsymbol{u}_{\text{opt}} = \arg\min_{\boldsymbol{u}} \left\{ L_e(\boldsymbol{x}_H, \boldsymbol{x}_H^r) + \sum_{t=1}^{H-1} L_r(\boldsymbol{x}_t, \boldsymbol{x}_t^r, \boldsymbol{u}_t) \right\} \tag{5a}$$

$$\text{s.t.} \quad \boldsymbol{x}_{t+1} = \boldsymbol{f}_{\text{dyn}}(\boldsymbol{x}_t, \boldsymbol{u}_t) \tag{5b}$$

$$\boldsymbol{x}_0 = \hat{\boldsymbol{x}}, \quad \boldsymbol{x}_t \in \mathcal{X} \tag{5c}$$

$$\boldsymbol{u}_{\text{lb}} \le \boldsymbol{u}_t \le \boldsymbol{u}_{\text{ub}} \tag{5d}$$

where $\boldsymbol{f}_{\text{dyn}}$ is the system dynamics, $\mathcal{X}$ the feasible state space, and $\boldsymbol{u}_{\text{lb}}, \boldsymbol{u}_{\text{ub}}$ the actuation bounds. Eq. (5a) uses terminal cost L_e and stage cost L_r that are quadratic functions of the errors, given by $\boldsymbol{e}^\top\boldsymbol{Q}\boldsymbol{e}, \quad \boldsymbol{e} \in \{\boldsymbol{e}_p, \boldsymbol{e}_R, \boldsymbol{e}_v, \boldsymbol{e}_\theta, \boldsymbol{e}_u\}$ where $\boldsymbol{Q}$ matrices are hand-tuned positive definite weights. Discretization is performed using a fourth-order Runge–Kutta scheme for stability.

In addition to producing control inputs, the MPC exposes a tracking cost L_{track} that quantifies how well the reference trajectory a can be followed under these constraints:

$$L_{\text{track}}(a) = \sum_{t=1}^{H} \left(\boldsymbol{e}_{p,t}^{\top} \boldsymbol{Q}_p \boldsymbol{e}_{p,t} + \boldsymbol{e}_{R,t}^{\top} \boldsymbol{Q}_R \boldsymbol{e}_{R,t} \right) \tag{6}$$

C. Embodiment-Aware Diffusion Guidance

We compute the gradient of L_{track} with respect to the reference trajectory, $\nabla_a L_{\text{track}}(a)$, which captures how sensitive the tracking error is to changes in the reference trajectory. In other words, it tells us how to nudge the reference trajectory a so that it becomes more trackable by the low-level controller.

As illustrated in Fig. 3, at inference time, we guide the conditional diffusion policy using this gradient feedback from the low-level controller. Let a^k denote the noisy reference trajectory sample at diffusion timestep $k \in \{K, \dots, 1\}$, conditioned on observation data $\boldsymbol{o}$. We use the standard DDIM [48] update step, given by:

$$a^{k-1} = a^k + \psi_k(\pi_\theta(a^k, t \mid \boldsymbol{o})), \tag{7}$$

where π_θ is the trained denoiser, and ψ_k is the DDIM update function for step k.

We incorporate gradient feedback from the tracking cost L_{track} defined in Eq. (2, 6) similarly to classifier-based guidance [49]. Specifically, we apply a guidance step to the trajectory sample toward feasible modes:

$$\tilde{a}^k = a^k - \lambda \cdot \bar{\omega}_k \cdot \nabla_{a^k} L_{\text{track}}(a^k), \tag{8}$$

where λ is a global guidance scale, and $\bar{\omega}_k \in (0, 1)$ is the guidance scheduler, equal to the cumulative noise schedule $\bar{\alpha}_k$. This makes guidance scale time-dependent: weak during early noisy steps and stronger during later denoising steps. We then use the nudged sample for denoising.

Algorithm 1 Embodiment-Aware DDIM Sampling

Initialize $a^K \sim \mathcal{N}(0, I)$ ▷ Start from noise
for $k = K, \dots, 1$ **do**
  $\tilde{a}^k \leftarrow a^k - \lambda \cdot \bar{\omega}_k \cdot \nabla_{a^k} L_{\text{track}}(a^k)$
  $a^{k-1} \leftarrow \tilde{a}^k + \psi_k(\pi_\theta(\tilde{a}^k, k \mid \boldsymbol{o}))$
end for
return a^0 ▷ Reference trajectory

The full procedure is summarized in Algorithm 1. The diffusion policy training remains independent of the embodiments used for deployment, but embodiment-specific controllers can inject real-time constraints and feasibility gradients. Thus, our method robustifies plug-and-play deployment across embodiments without retraining.

IV. Experimental Results

Our experiments aim to evaluate the extent to which embodiment-aware guidance improves the deployment of embodiment-agnostic visuomotor policies. We design both simulation and real-world studies to probe the embodiment gap and assess how well EADP addresses it. The key questions we investigate are the following:

1) How significant is the embodiment gap across different robots, and to what extent does EADP mitigate it?
2) Does EADP enable reliable transfer of UMI-trained policies to real-world UAMs?
3) Can UMI-on-Air generalize to unseen environments?

A. Simulation Experiments

To address our first question, we construct a controlled simulation benchmark in MuJoCo. Our setup allows us to systematically evaluate how an embodiment can affect policy execution across tasks.

We use motion capture on a UMI gripper to collect human demonstrations in simulation, mirroring the handheld demonstration process used in the real world. These demonstrations are used to train an *embodiment-agnostic* Diffusion Policy (DP), which serves as the base policy for comparison with EADP. We evaluate across four simulation environments, covering both long-horizon and precision tasks:

1) *Open-And-Retrieve*: Slide open a cabinet, pick up can, and place on top of cabinet. Can location is randomized.
2) *Peg-In-Hole*: Insert a 1cm peg into a 2cm square hole. 50s timeout if not pegged. Hole location is randomized.
3) *Rotate-Valve*: Rotating a valve to a specified orientation. Valve location is randomized.
4) *Pick-and-Place*: Lift a can and place it in a bowl. Object locations are randomized.

We deploy the trained policy across three embodiments, each reflecting different levels of control fidelity:

1) *Oracle*: A flying gripper that perfectly tracks the policy-generated trajectory. This provides an upper bound on achievable performance with no embodiment gap.
2) *UR10e*: A fixed-base 6-DoF manipulator, using an IK-based velocity-limited controller (§ III-B).
3) *UAM*: Aerial manipulator using the MPC controller (§ III-B). We consider two variants: (i) *UAM (no disturbance)*, and (ii) *UAM+Disturbance*, where we inject noise into the UAM base to simulate the $\sim$3 cm average tracking error observed on hardware when hovering near a still target. This allows us to test whether EADP can help compensate for real-world disturbances.

Fig. 6 reports success rates of DP and EADP across all tasks and embodiments. The gap between the *Oracle* and baseline DP serves as a natural measure of how "UMI-able" each embodiment is. As expected, the *UR10e* is close to *Oracle* performance, reflecting that tabletop manipulators with IK controllers can reliably track UMI policies. In contrast, the *UAM* exhibits a much larger gap—especially under disturbances—highlighting the difficulty of executing embodiment-agnostic trajectories on aerial systems.

EADP consistently reduces this embodiment gap. For *UR10e*, improvements are modest but noticeable on difficult tasks. For the *UAM*, EADP substantially boosts performance, recovering over 9% on average without disturbances and over 20% with disturbances. Even in the most constrained setting, EADP narrows the gap toward *Oracle*, confirming that embodiment-aware guidance enables policies to adapt trajectories to dynamic feasibility.

The *Open-and-Retrieve* task illustrates the challenges of long-horizon execution. Failures often occur when the gripper

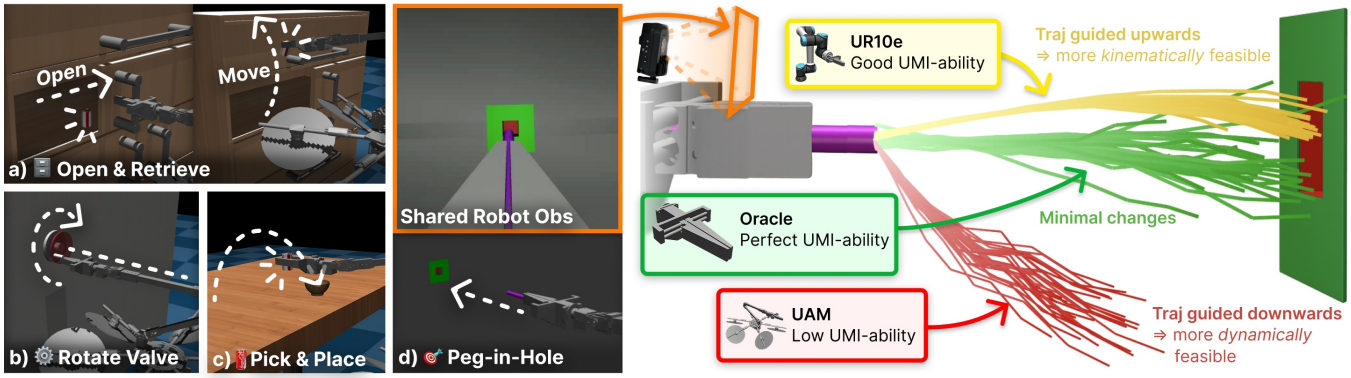

Fig. 5: **Policy Adaptation Across Embodiments.** Across four simulated tasks (1) and three embodiments (2), we observe that EADP can adapt the embodiment-agnostic diffusion policy to the deployment embodiments with varying "UMI-abilities". Visualizing 32 action samples across different embodiments for the same observation, we observe that **UR10e**'s trajectories is guided upwards to be **more kinematically feasible**, avoiding kinematic singularities. In contrast, the **UAM**'s trajectories are guided downwards to be **more dynamically feasible** due to perturbations along the $-Z$ direction.

jams on the cabinet door or when placing the can on top—*UR10e* slows down near its kinematic limits, while the *UAM* overshoots with momentum, causing collisions. Disturbances exacerbate these issues, pushing trajectories out-of-distribution (OOD). EADP mitigates many of these cases by steering trajectories toward safer, more in-distribution motions given by the policy.

In the *Peg-in-Hole* task, all embodiments succeed except the UAM with disturbances, where the hole is smaller than the average noise. This makes the task a stress test for disturbance robustness. EADP substantially improves reliability here, effectively rejecting infeasible pegging attempts under high noise and timing insertions when feasible, demonstrating that embodiment-aware guidance can even correct precision-sensitive behaviors (See Fig. 5).

The global guidance scale λ controls the trade-off between task-oriented trajectory generation and controller-feasible execution (Fig. 7) Without guidance ($\lambda = 0$), performance collapses under disturbances. As λ increases, success rates steadily improve. Excessively large λ over-constrains the denoising process, leading to conservative, OOD behaviors.

B. Real-world Experiments

We next address our second research question: *does EADP enable reliable transfer of UMI-trained policies to real-world UAMs?* To this end, we evaluate on three aerial manipulation tasks (Fig. 2) that span precision, robustness, and long-horizon execution, followed by a cross-environment generalization test. We conducted experiments using a fully actuated hexarotor drone with a 4 DoF manipulator and a gripper. A motion capture system provides the drone state, while the EE state is computed using forward kinematics. Fig. 8 summarizes results across all trials.

a) Peg-in-Hole.: We evaluate on a 4 cm hole with a 2 cm peg, with randomized starting positions and a 3 min timeout. While the baseline DP failed due dropped peg or a timeout, EADP succeeded on all five trials (5/5). By incorporating controller feedback, EADP generated trajectories that avoided premature release and improved timing during insertion.

b) Pick-and-Place (Lemon Harvesting).: In this task, the UAM must harvest a lemon from a randomized location and place it into a basket. EADP completed 4/5 trials successfully, with the only failure occurring when an unripe (green) lemon was selected. Notably, in this failure case the ripe yellow lemon was positioned near the edge of the camera's field of view from the initial state. Overall, EADP robustly handles aerial pick-and-place motions once a valid target is identified.

c) Lightbulb Insertion.: This long-horizon task requires threading a bulb into its socket until tight, followed by flipping the switch to confirm success. The task spans over 3 minutes of wall-clock time, underscoring the need for stability over extended horizons. EADP succeeded in all trials (3/3), demonstrating its ability to maintain precision and robustness throughout long-horizon tasks.

d) Cross-Environment Generalization.: Finally, we revisit the peg-in-hole task to probe our third research question: *can UMI-on-Air generalize to previously unseen environments?* We collect a dataset of demonstrations in varied real-world settings distinct from the test environment, then evaluate in a new environment with gradually increasing distractions across trials. With a 5 cm hole, EADP consistently located and aligned the peg, succeeding in 4/5 attempts. The only failure occurred when the drone collided with the hole's enclosure, leading to a miss.

V. Conclusion & Discussion

We introduced Embodiment-Aware Diffusion Policy (EADP), a framework for coupling embodiment-agnostic visuomotor policies with embodiment-specific controllers at inference time. Unlike standard UMI deployments that rely on one-way communication from high-level policy to low-level control, EADP enables controllers to provide gradient feedback on tracking feasibility, steering the diffusion sampling process toward trajectories that are dynamically feasible for the executing embodiment. This mechanism allows plug-and-play adaptation of UMI policies across

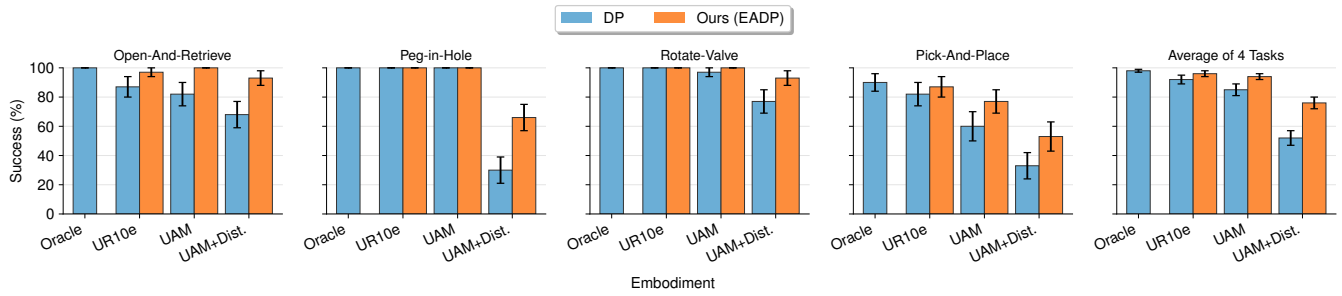

Fig. 6: **Simulation Results** for Diffusion Policy (DP) and Embodiment-Aware Diffusion Policy (EADP) on four tasks across four embodiments. EADP consistently outperforms DP, with larger gains in more constrained (less "UMI-able") embodiments.

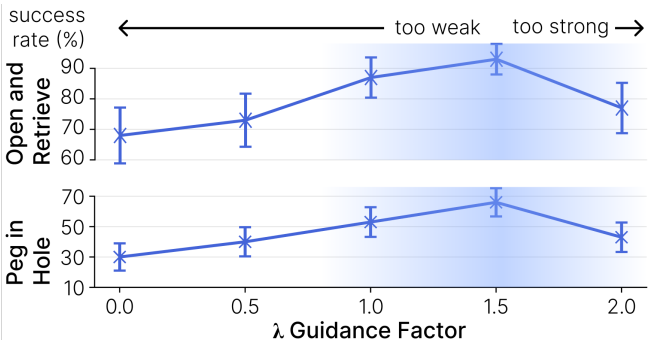

Fig. 7: **Guidance Ablation** for UAM+Disturbance.

diverse embodiments without retraining. Through large-scale simulation and real-world aerial manipulation experiments, we demonstrated that EADP consistently reduces the embodiment gap, especially in embodiments that are less "UMI-able."

While promising, our work leaves several directions for future research. First, the current system operates with a temporal gap between policy inference (around 1-2 Hz) and high-frequency control (50 Hz). This mismatch could be alleviated through streaming diffusion methods [50] or continuous guidance mechanisms that allow tighter integration between policy and controller. Second, while we demonstrate EADP with IK and MPC instantiations, the framework is not limited to analytical controllers. It can be naturally extended to learned or reinforcement learning–based controllers using learned dynamics models.

By bridging the gap between general, data-driven visuomotor policies and embodiment-specific feasibility, EADP represents a step toward scalable, universal manipulation. In doing so, it expands the practical deployment of UMI-style demonstrations from controlled lab settings to robots and environments that were previously beyond reach.

VI. Acknowledgment

This work was supported by the Robotics Institute Summer Scholars program. We thank Zeji Yi for his thoughtful discussions, and Yutong Wang and Mohammadreza Mousaei for their help with the experiments. This work was partially funded by NSF Award #2512805, #2143601, #2037101, and #2132519 and Toyota Research Institute. Guanya Shi holds concurrent appointments as an Assistant Professor at Carnegie Mellon University and as an Amazon Scholar. This paper describes work performed at Carnegie Mellon University and is not associated with Amazon.

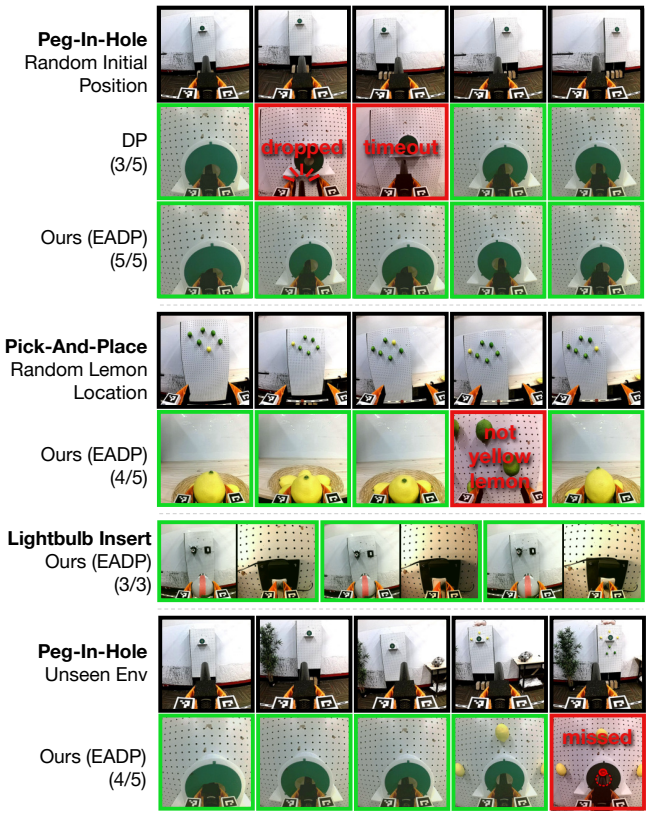

Fig. 8: **Real-World Results** for DP and EADP. Colored borders indicate **success** or **failure** for each trial.

References

[1] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song, "Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots," *arXiv preprint arXiv:2402.10329*, 2024.

[2] H. Ha, Y. Gao, Z. Fu, J. Tan, and S. Song, "UMI on legs: Making manipulation policies mobile with manipulation-centric whole-body controllers," in *Proceedings of the 2024 Conference on Robot Learning*, 2024.

[3] K. Bodie, M. Brunner, M. Pantic, S. Walser, P. Pfändler, U. Angst, R. Siegwart, and J. Nieto, "An omnidirectional aerial manipulation platform for contact-based inspection," *arXiv preprint arXiv:1905.03502*, 2019.

[4] A. S. Vempati, M. Kamel, N. Stilinovic, Q. Zhang, D. Reusser, I. Sa, J. Nieto, R. Siegwart, and P. Beardsley, "Paintcopter: An autonomous uav for spray painting on three-dimensional surfaces," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2862–2869, 2018.

[5] C. Ding, L. Lu, C. Wang, and C. Ding, "Design, sensing, and control of a novel uav platform for aerial drilling and screwing," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3176–3183, 2021.

[6] G. He, X. Guo, L. Tang, Y. Zhang, M. Mousaei, J. Xu, J. Geng, S. Scherer, and G. Shi, "Flying hand: End-effector-centric framework for versatile aerial manipulation teleoperation and policy learning," *arXiv preprint arXiv:2504.10334*, 2025.

[7] G. Shi, X. Shi, M. O'Connell, R. Yu, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung, "Neural lander: Stable drone landing control using learned dynamics," in *2019 international conference on robotics and automation (icra)*. IEEE, 2019, pp. 9784–9790.

[8] M. O'Connell, G. Shi, X. Shi, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung, "Neural-fly enables rapid learning for agile flight in strong winds," *Science Robotics*, vol. 7, no. 66, p. eabm6597, 2022.

[9] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, A. Casal, and A. Baader, "Force strategies for cooperative tasks in multiple mobile manipulation systems," in *Robotics Research: The Seventh International Symposium*. Springer, 1996, pp. 333–342.

[10] J. Chestnutt, M. Lau, G. Cheung, J. Kuffner, J. Hodgins, and T. Kanade, "Footstep planning for the honda asimo humanoid," in *Proceedings of the 2005 IEEE international conference on robotics and automation*. IEEE, 2005, pp. 629–634.

[11] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, "Optimization based full body control for the atlas robot," in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 120–127.

[12] E. Krotkov, D. Hackett, L. Jackel, M. Perschbacher, J. Pippine, J. Strauss, G. Pratt, and C. Orlowski, "The darpa robotics challenge finals: Results and perspectives," in *The DARPA robotics challenge finals: Humanoid robots to the rescue*. Springer, 2018, pp. 1–26.

[13] M. Bajracharya, J. Borders, R. Cheng, D. Helmick, L. Kaul, D. Kruse, J. Leichty, J. Ma, C. Matl, F. Michel *et al.*, "Demonstrating mobile manipulation in the wild: A metrics-driven approach," *arXiv preprint arXiv:2401.01474*, 2024.

[14] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," *arXiv preprint arXiv:2401.02117*, 2024.

[15] J. Yang, Z.-a. Cao, C. Deng, R. Antonova, S. Song, and J. Bohg, "Equibot: Sim (3)-equivariant diffusion policy for generalizable and data efficient learning," *arXiv preprint arXiv:2407.01479*, 2024.

[16] Y. Ze, Z. Chen, W. Wang, T. Chen, X. He, Y. Yuan, X. B. Peng, and J. Wu, "Generalizable humanoid manipulation with 3d diffusion policies," *arXiv preprint arXiv:2410.10803*, 2024.

[17] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. Kitani, C. Liu, and G. Shi, "Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning," *arXiv preprint arXiv:2406.08858*, 2024.

[18] Y. Jiang, R. Zhang, J. Wong, C. Wang, Y. Ze, H. Yin, C. Gokmen, S. Song, J. Wu, and L. Fei-Fei, "Behavior robot suite: Streamlining real-world whole-body manipulation for everyday household activities," *arXiv preprint arXiv:2503.05652*, 2025.

[19] F. Xia, C. Li, R. Martín-Martín, O. Litany, A. Toshev, and S. Savarese, "Relmogen: Integrating motion generation in reinforcement learning for mobile manipulation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4583–4590.

[20] J. Hu, P. Stone, and R. Martín-Martín, "Causal policy gradient for whole-body mobile manipulation," *arXiv preprint arXiv:2305.04866*, 2023.

[21] S. Uppal, A. Agarwal, H. Xiong, K. Shaw, and D. Pathak, "Spin: Simultaneous perception interaction and navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18 133–18 142.

[22] R. Mendonca, E. Panov, B. Bucher, J. Wang, and D. Pathak, "Continuously improving mobile manipulation with autonomous real-world rl," *arXiv preprint arXiv:2409.20568*, 2024.

[23] M. Liu, Z. Chen, X. Cheng, Y. Ji, R.-Z. Qiu, R. Yang, and X. Wang, "Visual whole-body control for legged loco-manipulation," *arXiv preprint arXiv:2403.16967*, 2024.

[24] X. Guo, G. He, M. Mousaei, J. Geng, G. Shi, and S. Scherer, "Aerial interaction with tactile sensing," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 1576–1582.

[25] C. Lanegger, M. Ruggia, M. Tognon, L. Ott, and R. Siegwart, "Aerial layouting: Design and control of a compliant and actuated end-effector for precise in-flight marking on ceilings," *Proceedings of Robotics: Science and System XVIII*, p. p073, 2022.

[26] X. Guo, G. He, J. Xu, M. Mousaei, J. Geng, S. Scherer, and G. Shi, "Flying calligrapher: Contact-aware motion and force planning and control for aerial manipulation," *arXiv preprint arXiv:2407.05587*, 2024.

[27] S. Ubellacker, A. Ray, J. M. Bern, J. Strader, and L. Carlone, "High-speed aerial grasping using a soft drone with onboard perception," *npj Robotics*, vol. 2, no. 1, p. 5, 2024. [Online]. Available: https://doi.org/10.1038/s44182-024-00012-1

[28] E. Bauer, M. Blöchlinger, P. Strauch, A. Raayatsanati, C. Curdin, and R. K. Katzschmann, "An open-source soft robotic platform for autonomous aerial manipulation in the wild," in *8th Annual Conference on Robot Learning*, 2024.

[29] M. Schuster, D. Bernstein, P. Reck, S. Hamaza, and M. Beitelschmidt, "Automated aerial screwing with a fully actuated aerial manipulator," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 3340–3347.

[30] M. Wang, Z. Chen, K. Guo, X. Yu, Y. Zhang, L. Guo, and W. Wang, "Millimeter-level pick and peg-in-hole task achieved by aerial manipulator," *IEEE Transactions on Robotics*, 2023.

[31] M. Brunner, G. Rizzi, M. Studiger, R. Siegwart, and M. Tognon, "A planning-and-control framework for aerial manipulation of articulated objects," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 689–10 696, 2022.

[32] D. Lee, H. Seo, I. Jang, S. J. Lee, and H. J. Kim, "Aerial manipulator pushing a movable structure using a dob-based robust controller," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 723–730, 2021.

[33] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu, "Rdt-1b: a diffusion foundation model for bimanual manipulation," *arXiv preprint arXiv:2410.07864*, 2024.

[34] J. Yang, C. Glossop, A. Bhorkar, D. Shah, Q. Vuong, C. Finn, D. Sadigh, and S. Levine, "Pushing the limits of cross-embodiment learning for manipulation and navigation," 2024.

[35] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-maron, M. Giménez, Y. Sulsky, J. Kay, J. T. Springenberg, T. Eccles, J. Bruce, A. Razavi, A. Edwards, N. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, and N. de Freitas, "A generalist agent," *Transactions on Machine Learning Research*, 2022.

[36] R.-Z. Qiu, S. Yang, X. Cheng, C. Chawla, J. Li, T. He, G. Yan, D. J. Yoon, R. Hoque, L. Paulsen *et al.*, "Humanoid policy˜ human policy," *arXiv preprint arXiv:2503.13441*, 2025.

[37] N. M. M. Shafiullah, A. Rai, H. Etukuru, Y. Liu, I. Misra, S. Chintala, and L. Pinto, "On bringing robots home," *arXiv preprint arXiv:2311.16098*, 2023.

[38] S. Song, A. Zeng, J. Lee, and T. Funkhouser, "Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4978–4985, 2020.

[39] S. Young, D. Gandhi, S. Tulsiani, A. Gupta, P. Abbeel, and L. Pinto, "Visual imitation made easy," in *Conference on Robot learning*. PMLR, 2021, pp. 1992–2005.

[40] M. Xu, H. Zhang, Y. Hou, Z. Xu, L. Fan, M. Veloso, and S. Song, "Dexumi: Using human hand as the universal manipulation interface for dexterous manipulation," *arXiv preprint arXiv:2505.21864*, 2025.

[41] T. Wang, R. Liao, J. Ba, and S. Fidler, "Nervenet: Learning structured policy with graph neural networks," in *International conference on learning representations*, 2018.

[42] W. Huang, I. Mordatch, and D. Pathak, "One policy to control them all: Shared modular policies for agent-agnostic control," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4455–4464.

[43] V. Kurin, M. Igl, T. Rocktäschel, W. Boehmer, and S. Whiteson, "My body is a cage: the role of morphology in graph-based incompatible control," *arXiv preprint arXiv:2010.01856*, 2020.

[44] A. Gupta, L. Fan, S. Ganguli, and L. Fei-Fei, "Metamorph: Learning universal controllers with transformers," *arXiv preprint arXiv:2203.11931*, 2022.

[45] C. Sferrazza, D.-M. Huang, F. Liu, J. Lee, and P. Abbeel, "Body transformer: Leveraging robot embodiment for policy learning," *arXiv preprint arXiv:2408.06316*, 2024.

[46] A. Patel and S. Song, "Get-zero: Graph embodiment transformer for zero-shot embodiment generalization," *arXiv preprint arXiv:2407.15002*, 2024.

[47] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference*

on Medical image computing and computer-assisted intervention. Springer, 2015, pp. 234–241.

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

[49] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021.

[50] S. Jiang, X. Fang, N. Roy, T. Lozano-Pérez, L. P. Kaelbling, and S. Ancha, "Streaming flow policy: Simplifying diffusion / flow-matching policies by treating action trajectories as flow trajectories," *arXiv preprint arXiv:2505.21851*, 2025.