π, But Make It Fly: Physics-Guided Transfer of VLA Models to Aerial Manipulation

Author Names Omitted for Anonymous Review. Paper-ID 1077

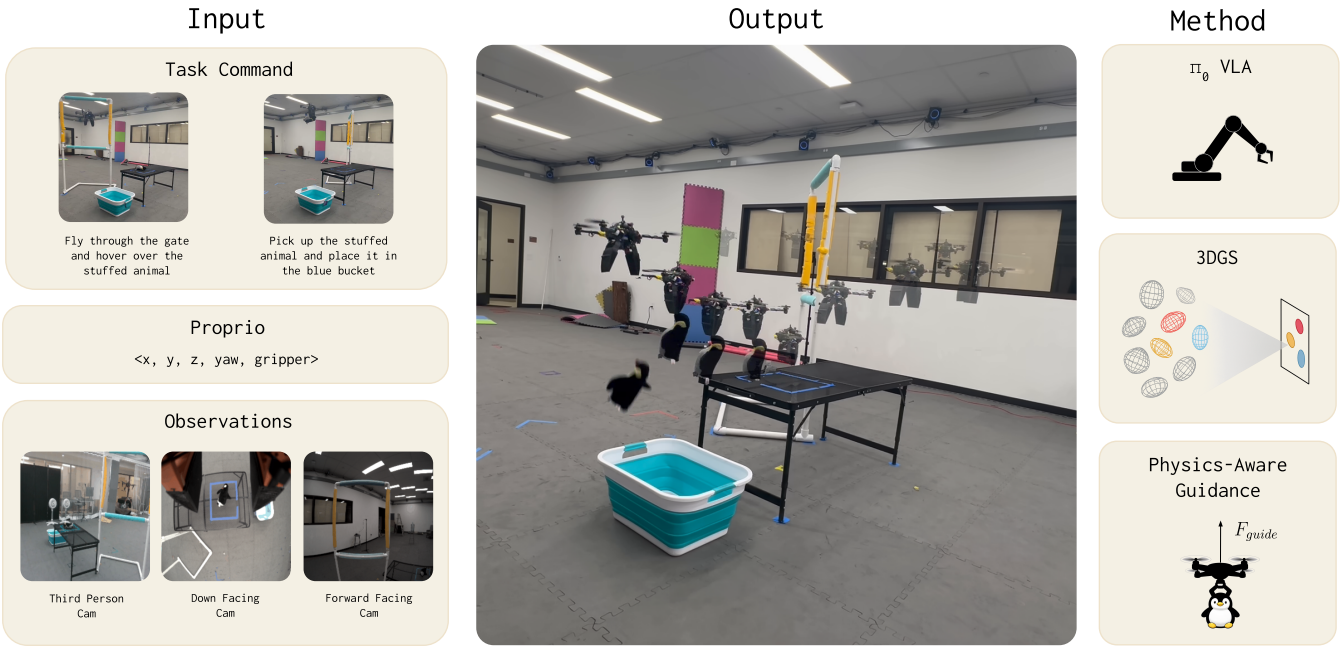

Fig. 1. **Overview of AirVLA:** Our method fine-tunes the π_0 vision-language-action model on a combination of teleoperated and 3D Gaussian Splatting synthetic data. **(Left)** The policy processes multimodal inputs including natural language commands, proprioception, and multi-view camera observations. **(Right)** To ensure robust flight during manipulation, we introduce a payload-aware guidance signal, F_{guide}, combined with real-time chunking. **(Center)** This enables AirVLA to execute novel, zero-shot compositional tasks, such as navigating through gates and manipulating objects.

***Abstract*—Vision-Language-Action (VLA) models such as π_0 have demonstrated remarkable generalization across diverse fixed-base manipulators. However, transferring these foundation models to aerial platforms remains an open challenge due to the fundamental mismatch between the quasi-static dynamics of fixed-base arms and the underactuated, highly dynamic nature of flight. In this work, we introduce AirVLA, a system that investigates the transferability of manipulation-pretrained VLAs to aerial pick-and-place tasks. We find that while visual representations transfer effectively, the specific control dynamics required for flight do not. To bridge this "dynamics gap" without retraining the foundation model, we introduce a Payload-Aware Guidance mechanism that injects payload constraints directly into the policy's flow-matching sampling process. To overcome data scarcity, we further utilize a Gaussian Splatting pipeline to synthesize navigation training data. We evaluate our method through a cumulative 460 real-world experiments which demonstrate that this synthetic data is a key enabler of performance, unlocking 100% success in navigation tasks where directly fine-tuning on teleoperation data alone attains 81% success. Our inference-time intervention, Payload-Aware Guidance, increases real-world pick-and-place task success from 23% to 50%. Finally, we evaluate the model on a long-horizon compositional task, achieving a 62% overall success rate. These results suggest that pre-trained manipulation VLAs, with appropriate data augmentation and physics-informed guidance, can transfer to aerial manipulation and navigation, as well as the composition of these tasks.**

***Index Terms*—Vision-Language-Action (VLA) Model, Vision-Language Model (VLM), Cross-Embodiment Transfer, Aerial Robotics, Unmanned Aerial Manipulators (UAMs)**

I. Introduction

Vision-Language-Action (VLA) foundation models have demonstrated remarkable cross-embodiment generalization, enabling robots with diverse morphologies to complete manipulation tasks from natural language instructions. Models such as RT-X [11], Octo [32], and π_0 [3] leverage large-scale multimodal pretraining across dozens of robot embodiments, learning representations that transfer across manipulators with different kinematics, workspaces, and end-effector designs. However, all such demonstrations share a fundamental constraint: they operate from stable, fixed or mobile bases in quasi-static regimes. This raises a central question: *can VLA policies transfer to aerial manipulators*, robots that couple perception, language understanding, and contact-rich interaction

with underactuated 6-DoF flight control? Such transfer would represent an extreme test of cross-embodiment learning, as aerial platforms differ fundamentally from ground robots in dynamics, control authority, and sensing characteristics.

Answering this question has both scientific and practical significance. Scientifically, aerial manipulation provides a stress test for the representations learned by VLA foundation models: if policies pretrained entirely on fixed-base manipulators can partially transfer to flying robots through fine-tuning, it suggests these models capture manipulation primitives that transcend specific embodiment constraints. Practically, language-conditioned aerial manipulation would enable applications previously beyond reach: delivering medical supplies in inaccessible terrain, clearing debris in collapsed structures, or manipulating infrastructure at height. If VLA policies could be adapted for such platforms, non-expert operators could issue high-level language commands while the policy handles low-level perception and control.

Several fundamental challenges distinguish aerial manipulation from the ground-robot scenarios that dominate VLA training data. First, quadrotors are underactuated: thrust and attitude are tightly coupled, and small errors in predicted actions can induce large deviations in pose. This failure mode rarely occurs in fixed-base manipulation. Second, onboard cameras experience large ego-motion, rapid viewpoint changes, motion blur, and scale variation that differ significantly from tabletop scenarios in most VLA datasets [23, 11]. Finally, contact with objects induces payload changes [5, 19] and aerodynamic disturbances [35, 33] that violate quasi-static assumptions implicit in most manipulation policies.

While recent work has begun exploring language-conditioned aerial systems and cross-embodiment transfer to drones, no prior work has systematically investigated whether manipulation-pretrained VLA foundation models can transfer to aerial platforms. Concurrent efforts take fundamentally different approaches: UMI-on-Air [17] trains embodiment-agnostic diffusion policies from scratch on handheld demonstrations and introduces inference-time guidance to compensate for aerial dynamics, but does not leverage or evaluate VLA foundation models. SINGER [1] and GRaD-Nav++ [8] demonstrate language-conditioned drone navigation using learned visuomotor policies, but focus exclusively on navigation without physical manipulation or contact. More broadly, while cross-embodiment datasets such as Open X-Embodiment [11] include dozens of manipulators, they contain no aerial platforms or underactuated flight dynamics, leaving the transferability of VLA representations to aerial manipulation unexplored.

To address these gaps, we introduce *AirVLA*, the first systematic investigation of whether manipulation-pretrained VLA foundation models can transfer to aerial manipulators through fine-tuning. Our system consists of a ModalAI Starling 2 Max quadrotor equipped with a lightweight compliant gripper derived from the Universal Manipulation Interface (UMI) [10, 18, 17], forward- and downward-facing onboard cameras, and a ROS-based control stack that treats the drone as a "flying end-effector." We construct a diagnostic task suite spanning (i) *pick-and-place* manipulation collected via human teleoperation, (ii) *navigation* tasks with human teleoperation data and synthetic training trajectories generated using model predictive control (MPC) in Gaussian Splatting reconstructions, and (iii) *compositional* tasks that chain both behaviors. This design enables us to isolate which aspects of manipulation skill transfer to aerial platforms and which fail.

We directly fine-tune the foundation-scale VLA policy π_0 (pretrained on large manipulation datasets containing no aerial platforms) on our aerial manipulation data. Our experiments reveal partial but significant skill transfer: the fine-tuned policy learns stable hovering and coarse manipulation behaviors, demonstrating that foundation models capture some manipulation primitives that generalize to flight. However, standard fine-tuning fails at task completion due to sensitivity to payload dynamics and explicit obstacle navigation limitations. To address these specific failure modes, we introduce domain-adapted components: (i) a physics-aware low-level controller that wraps VLA outputs to enforce dynamically feasible commands, and (ii) synthetic navigation augmentation that enriches the training distribution. While these adaptations improve performance on navigation and stabilization during manipulation, fundamental limitations remain for long-horizon compositional manipulation and out-of-distribution adaptation, highlighting important open challenges in extending VLA models to underactuated, dynamic embodiments.

In summary, this paper makes the following contributions:

- We present AirVLA, the first demonstration of a pretrained VLA fine-tuned for an aerial manipulation platform.
- We manually collect a dataset of 270 aerial manipulation and navigation teleop-demos, to be open-sourced.
- We synthesize 50 3DGS-based corrective navigation examples to help supervise VLA fine-tuning, showing a 20% increase in navigation task success compared to fine-tuning with teleop only.
- We introduce a physics-informed guidance within real-time-chunking to adapt at runtime to the mass of a grasped object, improving task success by $10\% - 40\%$ in pick and place tasks.

II. Related Work

A. Vision-Language-Action Policies for Manipulation

Vision-language-action (VLA) policies leverage large-scale multimodal pretraining to enable general-purpose robot control from natural language instructions. Early approaches such as RT-1 [6] and RT-2 [7] demonstrated that vision-language models pretrained on internet-scale data can be fine-tuned for robotic manipulation, mapping visual observations and language to action sequences. Subsequent work has scaled this paradigm through cross-embodiment training: the Open X-Embodiment dataset [11] aggregates demonstrations from dozens of robot morphologies, enabling models like RT-X [11] to generalize across diverse manipulators. More recent robot

foundation models [2, 24] such as Octo [32] and π_0 [3] incorporate flow-matching and diffusion architectures, achieving state-of-the-art performance on manipulation benchmarks spanning tabletop tasks, mobile manipulation, and dexterous control.

These successes demonstrate that VLA policies can learn manipulation primitives that transfer across embodiments with different kinematics, workspaces, and end-effector designs. However, essentially all demonstrations in existing cross-embodiment datasets operate from stable bases (fixed or mobile) in quasi-static regimes. No prior work has investigated whether these learned representations transfer to aerial platforms, where underactuated flight dynamics, large ego-motion, and contact-induced disturbances fundamentally differ from ground-robot scenarios. Our work provides the first systematic evaluation of this transfer gap.

B. Vision-Language and Vision-Language-Action Models for UAV Navigation

Vision-language models have recently been applied to drone navigation, typically for high-level semantic reasoning rather than low-level visuomotor control. VLMaps [21] constructs semantic maps for language-conditioned robot navigation but act as a perception and planning front-end rather than an end-to-end policy. SEEK [15] and VISTA [31] demonstrate uncertainty-aware exploration from semantic goals on quadrupeds and drones, but are limited to object localization and do not extend to interaction or contact. More recently, SINGER [1] introduces an end-to-end visuomotor policy for language-conditioned drone navigation trained on synthetic trajectories in Gaussian Splatting environments, achieving onboard inference through semantic image preprocessing with CLIPSeg [30]. GRaD-Nav++ [8] extends this approach with differentiable dynamics in Gaussian Splatting scenes, enabling multi-task generalization across navigation behaviors. Both methods demonstrate impressive sim-to-real transfer for navigation tasks.

In contrast to SINGER and GRaD-Nav++, which focus on language-conditioned navigation without contact and use Gaussian Splatting primarily as a training environment, our work targets aerial manipulation with an onboard gripper, uses Gaussian Splatting as a source of synthetic demonstration data for a specific navigation subtask, and explicitly studies how a foundation VLA policy can be leveraged when navigation and manipulation must be composed in a single language-specified behavior.

C. Aerial Manipulation with Learned Policies

Unmanned aerial manipulation (UAM) combines multirotor flight with contact-rich interaction, introducing challenges such as underactuated dynamics, aerodynamic disturbances near surfaces, and strict payload limits. Classical systems demonstrate contact-based tasks including surface inspection [39, 5], painting [40, 16, 27], drilling [13], and object grasping [14, 43], typically using carefully engineered controllers tailored to specific scenarios. Recent work has explored learning-based approaches: model-free reinforcement learning for aerial manipulation [12], one-shot learning for autonomous grasping [45], and image-based visual servoing [38, 26]. However, these methods either lack language grounding or are limited to single tasks without generalization across manipulation primitives.

Most relevant to our work, UMI-on-Air [17] addresses the challenge of transferring manipulation policies to aerial platforms by training diffusion policies on handheld UMI demonstrations and introducing embodiment-aware guidance: gradient feedback from a low-level MPC controller steers trajectory generation toward dynamically feasible actions at inference time. This approach achieves impressive results on high-precision aerial manipulation tasks.

Compared to UMI-on-Air, which uses a diffusion policy trained from handheld UMI demonstrations and then introduces embodiment-aware guidance at inference time via MPC gradients, AirVLA directly fine-tunes a large VLA foundation model (π_0) on data collected from the aerial manipulator itself and uses the resulting policy as trained during deployment. This design lets us isolate and study representational generalization limits of existing VLA models to aerial manipulation.

More broadly, while prior aerial manipulation work either (i) relies on task-specific controllers without language grounding or (ii) uses cross-embodiment guidance to make existing policies feasible on drones, none of these efforts evaluate a generalist VLA manipulation model on a unified suite of pick-and-place, navigation, and compositional aerial tasks the way AirVLA does.

III. Method

A. System Overview

AirVLA is a vision-language-action system for aerial manipulation that transfers manipulation capabilities from foundation models pretrained on fixed-base robot arms to an underactuated quadrotor platform. The system takes as input RGB images from multiple viewpoints and a natural language task description, and outputs relative end-effector pose commands executed by a low-level flight controller.

The key challenge in this transfer is the mismatch between the quasi-static regimes of tabletop manipulation and aerial manipulation, where the platform must continuously stabilize against gravity while simultaneously executing precise gripper motions. Grasping an object introduces a step change in effective mass that, if uncompensated, causes the drone to sag and potentially fail the task.

Our approach addresses this challenge through two main contributions: (1) a physics-aware guidance mechanism that augments the pretrained policy's action generation process with payload-aware vertical compensation at inference time, and (2) a Gaussian-splatting data pipeline that enables efficient collection and synthesis of diverse training trajectories from a small number of seed flights. Together, these allow a VLA model pretrained on large-scale robot manipulation data to perform aerial pick-and-place and navigation with minimal drone-specific fine-tuning.

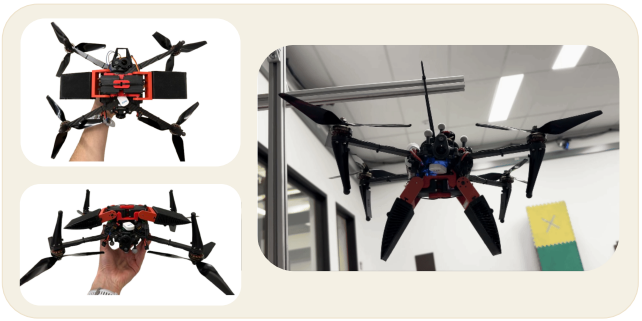

Fig. 2. Custom gripper installed on the Starling 2 Max drone. The design prioritizes low weight for extended flight time while maintaining sufficient grip strength for the target objects.

B. Hardware

The system integrates the ModalAI Starling 2 Max drone with a customized Universal Manipulation Interface (UMI) gripper [10] and multiple cameras to enable autonomous aerial manipulation. The Starling 2 Max is powered by the VOXL 2 companion computer (Qualcomm QRB5165). The customized UMI-style gripper is attached to the underside of the drone, enabling dynamic grasping. The system integrates three external cameras and two onboard cameras (downward and forward-facing), providing RGB images at 5Hz.

1) Gripper Design: We showcase the design of our gripper in Fig. 2. The gripper is designed to be built cheaply without specialized tools. The frame is entirely 3D printed, with two hobby-grade servos slotting in without screws. The fingers are adapted from the UMI gripper [10], facilitating direct comparison with arm-based policies trained with similar end-effector geometry.

2) Observation and Action Spaces: The observation space consists of RGB images from three cameras at $[256 \times 256]$ resolution, the drone's estimated pose from a motion capture system, and gripper aperture. The action space, consisting of the drone position and yaw, is represented as an action chunk $A \in \mathbb{R}^{H\times D}$ (i.e., end-effector 4 DoF delta poses and gripper commands D over horizon H). Actions are generated at 10 Hz and executed by the PX4 flight controller via position setpoints.

C. Policy Architecture:

We build on π_0 [3], a vision-language-action model that represents the conditional action distribution using a flow-matching (continuous-time) generative model [28]. Given an observation o (images, proprioception, language), π_0 defines a velocity field $v_\theta(x_\tau, o, \tau)$ over latent action chunks $x_\tau \in \mathbb{R}^{H\times D}$ and diffusion/flow time $\tau \in [0, 1]$. Sampling draws $x_0 \sim \mathcal{N}(0, I)$ and integrates the ODE

$$\frac{dx_\tau}{d\tau} = v_\theta(x_\tau, o, \tau) \tag{1}$$

to obtain the generated action chunk $A = x_1$.

For real-time execution, we employ Real-Time Chunking (RTC) [4], which enables asynchronous inference by *freezing* the prefix of the next chunk that will execute before inference completes, and *inpainting* the remaining suffix conditioned on the frozen prefix. Concretely, RTC defines a soft temporal mask over the horizon that blends previously committed actions with newly generated actions to avoid discontinuities at chunk boundaries.

D. Physics-Aware Guidance for Action Generation

RTC shows that inference-time steering can be implemented by modifying the velocity field during sampling [4]. We generalize this idea by introducing a loss $\Phi(A; o)$ over the *denoised* action chunk $A = x_1$, and adding a gradient correction term derived from $\nabla_A \Phi$ to the base velocity field, analogous in spirit to gradient-based guidance methods in generative modeling [20, 36]. Intuitively, we seek samples that both (i) have high probability under the base policy and (ii) minimize the guidance loss:

$$p_{\text{guid}}(A \mid o) \propto p_\theta(A \mid o)\, \exp\big(-\Phi(A; o)\big). \tag{2}$$

This formulation seeks a 'sweet spot' between the policy's priors and physical constraints. It biases the sampling process toward actions that are high-probability under the VLA (preserving learned manipulation skills) while simultaneously minimizing the cost Φ (enforcing flight feasibility), effectively steering the drone at runtime without retraining.

Operationally, at intermediate time τ, we compute the model's current prediction of the terminal action chunk $\hat{A}_\theta(x_\tau, o, \tau) \approx x_1$ (e.g., via the model's internal denoised estimate), evaluate the gradient $\nabla_A \Phi(\hat{A}_\theta; o)$, and map it back to a correction in latent space through a vector–Jacobian product $\xi := \Big(\nabla_{x_\tau} \hat{A}_\theta(x_\tau, o, \tau)\Big)^\top \nabla_A \Phi\Big(\hat{A}_\theta(x_\tau, o, \tau); o\Big)$. This yields the guided velocity field

$$v_{\text{guid}}(x_\tau, o, \tau) = v_\theta(x_\tau, o, \tau) - s(\tau)\xi. \tag{3}$$

where $s(\tau)$ is a scalar guidance schedule. Multiple loss terms compose additively because $-\log p_{\text{guid}}$ adds losses, so RTC-style continuity objectives and task-specific physics objectives can be applied simultaneously without changing the pretrained model weights.

1) General Tracking-Error Guidance: Suppose we have a reference trajectory $A^{\text{des}}(o) \in \mathbb{R}^{H\times D}$ derived from the current observation. We define a tracking loss

$$\Phi_{\text{track}}(A; o) = \frac{1}{2} \sum_{t=0}^{H-1} \sum_{d=1}^{D} \lambda_d\, w_t \left(A_{t,d} - A^{\text{des}}_{t,d}(o)\right)^2, \tag{4}$$

with per-dimension strengths $\lambda_d \geq 0$ and temporal weights $w_t \geq 0$. The temporal weights follow the soft masking schedule from RTC [4]: $w_t = 1$ for frozen prefix actions, the weights then exponentially decay for the intermediate region and are zero for freshly generated actions. This ensures the guidance strongly enforces continuity with committed actions while allowing freedom for new generation. The corresponding gradient correction is derived from

$$\frac{\partial \Phi_{\text{track}}}{\partial A_{t,d}} = \lambda_d\, w_t \left(A_{t,d} - A^{\text{des}}_{t,d}(o)\right), \tag{5}$$

which pulls the denoised chunk toward the reference trajectory in each dimension proportionally to λ_d.

2) Payload-Aware Vertical Guidance: In aerial manipulation, the dominant disturbance during grasping is an effective mass increase that manifests as vertical sag under load. Rather than modeling full 6-DOF dynamics, we instantiate (4) only on the altitude-related action dimension d_z:

$$\Phi_{\text{payload}}(A; o, A_{t-1}) = \frac{\lambda_z}{2}\,\alpha(o, A_{t-1}) \sum_{t=0}^{H-1} w_t \left(z_t(A) - z_{\text{des}}(o)\right)^2, \tag{6}$$

where $z_t(A)$ denotes the z-component of the action at timestep t, and

$$z_{\text{des}}(o) = z_{\text{curr}}(o) + \Delta z \tag{7}$$

biases the drone toward slightly higher altitude under load ($\Delta z > 0$). Here $z_{\text{curr}}(o)$ comes from a motion capture system, and Δz is a tuned offset capturing the expected sag for typical payloads.

The payload confidence $\alpha(o, A_{t-1}) \in [0, 1]$ is computed from (i) the previously executed action chunk A_{t-1} and (ii) the current measured gripper aperture in o. Concretely, we compute a smooth, bounded payload confidence by combining recent gripper command intent with the current measured aperture. Let $u_t \in [-1, 1]$ denote the gripper command at timestep t (with $+1$ corresponding to "close"), and let $g(o) \in [0, 1]$ denote the measured aperture (normalized so that larger values are more open). From the previously executed chunk A_{t-1}, we form continuous close/open intents using the last K commands,

$$\begin{aligned} c_{\text{intent}} &= \frac{1}{K} \sum_{i=H-K}^{H-1} \text{clip}\left(\frac{u_i + 1}{2},\, 0,\, 1\right), \\ o_{\text{intent}} &= \frac{1}{K} \sum_{i=H-K}^{H-1} \text{clip}\left(1 - \frac{u_i + 1}{2},\, 0,\, 1\right), \end{aligned} \tag{8}$$

and compute soft aperture-based scores $c_{\text{meas}}, o_{\text{meas}} \in [0, 1]$ from $g(o)$ (higher when the gripper is closed/open, respectively). We then define an "open" gate and payload confidence as

$$\begin{aligned} o_{\text{flag}} &= \text{clip}\left(\tfrac{1}{2} o_{\text{intent}} + \tfrac{1}{2} o_{\text{meas}},\, 0,\, 1\right), \\ \alpha(o, A_{t-1}) &= \text{clip}\left((1 - o_{\text{flag}})\, c_{\text{intent}}\, c_{\text{meas}},\, 0,\, 1\right). \end{aligned} \tag{9}$$

When $\alpha \approx 0$, the loss vanishes and sampling reduces to vanilla RTC; when $\alpha \approx 1$, the sampler prefers chunks whose vertical component is biased toward z_{des}, compensating for payload. This can be viewed as mode-dependent feedforward (gain scheduling / gravity compensation), but injected *inside* the generative policy's sampling dynamics by adding the gradient correction term derived from $\nabla_A \Phi_{\text{payload}}$ via (3).

E. Gaussian Splat Data Pipeline

Collecting aerial manipulation demonstrations is time-consuming and requires skilled pilots. To efficiently generate diverse training data, we adopt a "Flying in Gaussian Splats"-style approach that couples photorealistic Gaussian-splatting reconstructions with a lightweight drone dynamics model to synthesize large volumes of training data from a small set of seed demonstrations [29]. Concretely, our pipeline proceeds by first reconstructing the static environment as a Gaussian Splat and isolating the gripper visuals to prevent observation bias. We then couple a drone dynamics model with these assets to synthesize diverse, physics-feasible training trajectories that cover both nominal navigation and recovery behaviors.

1) Scene Reconstruction: We reconstruct each scene from short walk-throughs captured with the drone's forward-facing camera. This results in a set of metrically scaled poses for each image, which are used to train a 3D Gaussian splatting model [22] using Nerfstudio [37, 42]. The resulting model $\mathcal{GS}_\phi$ renders photorealistic images from arbitrary camera poses in the captured region. Given a camera pose (p, q), the rendered image is $I = \mathcal{GS}_\phi(p, q)$.

2) Gripper Segmentation and Compositing: The downward-facing camera provides critical visual information for manipulation, but the gripper is persistently visible in its field of view. Including raw downward facing images in the synthetic training data from the Gaussian splatting model would introduce an observation bias into the policy that would result in unwanted behavior. Thus, we explicitly treat the gripper as a separate foreground layer and composite it onto renders from a gripper-free scene model. Concretely, for each downward-facing training frame I_{down}, we compute a gripper mask M_{grip} using Segment Anything (SAM) [25, 34] with a fixed bounding box prompt corresponding to the gripper's known image-space location. We then extract a masked gripper patch

$$G = I_{\text{down}} \odot M_{\text{grip}}, \tag{10}$$

to form a small library $\{G_a\}$ of representative gripper appearances.

We train the Gaussian splatting model $\mathcal{GS}_\phi$ on images from cameras where the gripper is not visible (namely, the forward-facing views). At synthesis time, we render the clean scene from $\mathcal{GS}_\phi$ for the downward-facing viewpoint and composite the gripper foreground:

$$I_{\text{down}}^{\text{synth}} = (1 - M_{\text{grip}}) \odot \mathcal{GS}_\phi(p, q) \;+\; M_{\text{grip}} \odot G_{a(t)}. \tag{11}$$

This yields a downward-view image whose background is fully determined by the scene model, while the gripper appearance is consistent with the commanded aperture $a(t)$.

3) Drone Dynamics Model: Following [29], we simulate drone motion with a semi-kinematic state $x = (p^W, v^W, q_B^W)$ comprising position, velocity, and orientation quaternion. Controls are normalized thrust f_{th} and body angular velocity ω^B. The dynamics are

$$\dot{p}^W = v^W, \tag{12}$$

$$\dot{v}^W = g\, e_3^W + \frac{k_{\text{th}}}{m}\, f_{\text{th}}\, R(q_B^W)\, e_3^B, \tag{13}$$

$$\dot{q}_B^W = \tfrac{1}{2}\, \Omega(\omega^B)\, q_B^W, \tag{14}$$

where g is gravitational acceleration, e_3^W, e_3^B are the z-axis unit vectors in world/body frames, $R(\cdot)$ converts quaternion

to rotation, and (k_{th}, m) are thrust coefficient and mass. We forward-integrate using ACADOS [41] and render images using the body-to-camera transform T_C^B.

4) Domain-Randomized Data Synthesis: To enable the policy to recover from dangerous states near the obstacle, we additionally synthesize recovery trajectories by randomizing task geometry, similar to FiGS [29]. For each nominal trajectory (X^d, U^d), we generate N_s randomized rollouts by sampling an initial state perturbation $x_0^s \sim \mathcal{U}(x_0^d - \Delta x,\, x_0^d + \Delta x)$, then simulating the resulting trajectory rendering multi-view images from the Gaussian splat scene model $\mathcal{GS}_\phi$.

We additionally randomize intermediate waypoints to increase diversity and induce recovery behaviors. For the navigation task, we randomize the terminal hover location to lie above the goal by sampling a height offset uniformly in $[1.0, 1.5]$ meters. Let p_{obj} denote the object goal position in the scene and let $h \sim \mathcal{U}(1.0, 1.5)$ m. We set the goal position as

$$p_{\text{goal}} = p_{\text{obj}} + [0, 0, h]^\top. \tag{15}$$

To diversify gate exits, we randomize the post-gate waypoint within a ball of diameter 0.25 m centered at a nominal after-gate position waypoint p^d_{after}. With radius $r = 0.125$ m and $\delta \sim \mathcal{U}(\mathbb{B}(0, r))$ where $\mathbb{B}(0, r) = \{\delta \in \mathbb{R}^3 : \|\delta\|_2 \le r\}$, we define

$$p_{\text{after}} = p^d_{\text{after}} + \delta. \tag{16}$$

Finally, we perturb trajectories to elicit recovery behavior by inserting an additional waypoint between the start and the gate that forces the drone to pass near one of four gate extremities, namely top, bottom, left, or right. We sample a side $s \sim \text{Unif}(\{\text{top}, \text{bottom}, \text{left}, \text{right}\})$ and define a side-specific waypoint

$$p_{\text{wp}} = \bar{p}_s + b\, n_s, \tag{17}$$

where $\bar{p}_s$ is a reference point near the corresponding gate boundary, n_s is the offset direction away from the gate frame, and $b > 0$ is a fixed geometry buffer chosen to ensure collision-free clearance.

After sampling these randomized waypoint targets, we generate controls by tracking the waypoint-conditioned plan under the drone dynamics, producing the rollout (X^s, U^s). Figure 3 depicts nominal and corrective synthetic trajectories for both the left gate and right gate scenarios. We render images along X^s from $\mathcal{GS}_\phi$ using the body-to-camera transform T_C^B, and composite the gripper for downward-facing views. This procedure yields a large set of physically and geometrically plausible trajectories that cover both nominal executions and off-nominal approaches requiring recovery near the gate.

IV. Experiments

To evaluate our methods, we define a task-specific score rubric, report quantitative results under that rubric, and organize experiments to isolate the contribution of pretraining transfer, inference-time stabilization, and data augmentation.

Through our experiments, we seek to study the following questions:

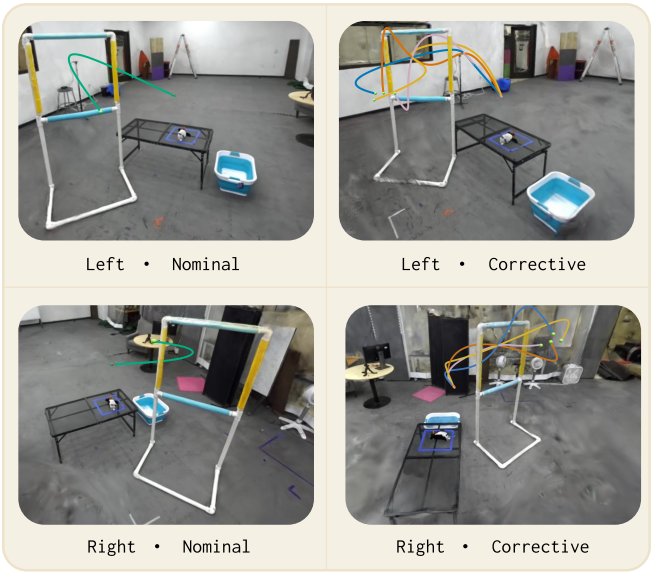

Fig. 3. Examples of synthetic trajectories used to generate training data. Nominal rollouts and corrective rollouts overlay sampled trajectories in the gate-manipulation scene.

1) **Transfer:** How much manipulation-pretrained VLA capability transfers to an underactuated aerial manipulator after fine-tuning?
2) **Inference-time control:** Can real-time chunking (RTC) and payload-aware guidance reduce the sensitivity of aerial execution to chunk discontinuities and payload disturbances?
3) **Data synthesis:** Does Gaussian-splat-based synthetic navigation augmentation improve real-world navigation performance?
4) **Compositionality:** Can a single policy reliably compose navigation and grasping into a multi-stage task (navigate-then-grasp)?

A. Task suite

We evaluate using the following task prompts:

- **Penguin Grasp (Pick-and-Place):** pick up the stuffed animal and put it in the blue bin.
- **Gate Navigation:** fly through the gate and hover over the stuffed animal.
- **Compositional (Navigate-then-Grasp):** fly through the gate and hover over the stuffed animal and then pick up the stuffed animal and put it in the blue bin.

The penguin grasp task evaluates the model's ability to transfer manipulation skills from pretraining to a drone embodiment. To ensure robustness, we vary the object's position within a box during both data collection and evaluation. The gate navigation task tests drone-specific navigation requiring explicit obstacle avoidance. By using two different gate positions (left and right), we force the policy to localize the gate prior to navigation. Finally, to test the model's ability to handle novel, composite tasks in a zero-shot manner, we fine-tune the policy

TABLE I
SINGLE TASK PERFORMANCE BENCHMARKS (%). EACH METHOD WAS EVALUATED ACROSS TWENTY TRIALS PER TASK.

	Penguin Grasp		**Navigation (Non-Synthetic)**		**Navigation (Synthetic)**	
Method	Pick	Place	Gate	Hover	Gate	Hover
π_0 naive	50.0	0.0	50.0	60.0	45.0	100.0
π_0 + RTC	85.0	23.5	80.0	81.2	95.0	100.0
π_0 + RTC w/ payload-aware guidance (ours)	100.0	50.0	–	–	–	–
ACT	0.0	0.0	0.0	0.0	0.0	0.0
Diffusion Policy	10.0	0.0	15.0	0.0	0.0	0.0

on the combined datasets and evaluate it using a combined prompt unseen during training.

B. Evaluation rubric and protocol

For each task, we define a rubric that measures progress toward completion. In our setting, the primary metric is binary task success, and we additionally record a diagnostic failure taxonomy to attribute failures for qualitative analysis:

- **Penguin Grasp:** a two stage task with success markers triggered after (1) picking up the stuffed animal and (2) placing it in the bin. Failure modes include grasping followed by immediate drops or crashes as well as failure to pick.
- **Gate Navigation:** a two stage task with success markers triggered after (1) navigating through the gate and (2) holding a hover over the stuffed animal. Failure modes include passing through the gate but not hovering over the penguin, and crashing or failing to pass through the gate.
- **Compositional Navigate and Grasp:** a four stage task combining the success and failure markers for the above with an additional failure marker of *incorrect subtask order* (attempted grasp before passing the gate).

C. Methods compared

We compare the following methods:

- π_0 **naive:** roll out action chunks, pausing for inference at the end of each action chunk.
- π_0 **+ RTC:** real-time chunking that re-samples suffix actions to avoid discontinuities [4].
- π_0 **+ RTC + payload-aware guidance (ours):** inference-time steering implemented via a gradient correction term applied inside the sampler dynamics [4].
- **ACT** [44] and **Diffusion Policy** [9]

For the navigation and compositional tasks, we additionally evaluate **synthetic** and **non-synthetic** variants, corresponding to whether the policy was trained with synthetic data augmentation.

D. Quantitative results

Each of the above methods are evaluated across twenty trials per task for a cumulative 460 flight trials. We omit evaluations of the payload-aware guidance method on pure navigation tasks, as the guidance method is identical to RTC when objects are not being manipulated. We present the results of these evaluations in Tables I and II. The reported success rates are conditioned on the number of trials that succeeded in the prior stage of the task. For example, the place success rate is conditioned on the number of trials that succeeded in picking the object.

To evaluate generalization to novel objects, we conduct 10 additional pick-and-place trials, replacing the objects used during training with previously unseen ones. Furthermore, to assess spatial robustness, we vary the gate location across three regions (front, left, and right). We conduct five trials per region, shifting and rotating the gate within each area. These out-of-distribution results are detailed in Table III.

E. Analysis

Inference-time structure is critical for aerial pick-and-place. Naive fine-tuning is insufficient for full pick-and-place success, attaining 0%, with failures dominated by missed grasps and post-contact disturbances. RTC improves success (up to 23.5%) by stabilizing execution across chunk boundaries, and payload-aware guidance further improves success (up to 50%), consistent with the need to compensate for payload-induced altitude sag and other underactuated dynamics.

RTC substantially improves gate navigation. In the non-synthetic gate navigation trials, RTC increases gate traversal success from 50% to 80% and from 45% to 95% in the synthetic gate navigation trials. This is likely because RTC reduces crash/failed-pass outcomes, supporting the hypothesis that re-planning at runtime mitigates compounding errors during aggressive motion.

Synthetic augmentation helps most when paired with RTC. In the synthetic setting, naive performance is asymmetric across gates while π_0+RTC achieves 95% success at the gate traversal component of the task. This suggests that synthetic augmentation can expand coverage of approaches/recoveries, but reliable deployment still benefits from inference-time stabilization. Qualitatively, we observed that the policy would periodically fall back on "synthetic modes" that replay the GS-generated trajectories, ensuring successful gate traversal.

Compositional tasks reveal an ability to generalize from atomic tasks. Our method shows a strong ability to generalize to compositional tasks in a zero-shot matter attaining a 62% overall conditioned success rate. The compositional task evaluations further highlight the importance of closed-loop policy

TABLE II
COMPOSITIONAL NAVIGATE-THEN-GRASP SUCCESS (%). EACH METHOD WAS EVALUATED ACROSS TWENTY TRIALS PER TASK.

Data Setting	Method	Gate	Hover	Pick	Place
No Synthetic	π_0 naive	35.0	85.7	42.9	0.0
	π_0 + RTC	80.0	100.0	81.2	15.4
	π_0 + RTC w/ payload-aware guidance (ours)	70.0	100.0	100.0	35.7
	ACT	0.0	0.0	0.0	0.0
	Diffusion Policy	5.0	0.0	0.0	0.0
Synthetic	π_0 naive	70.0	85.7	25.0	0.0
	π_0 + RTC	95.0	94.7	83.3	20.0
	π_0 + RTC w/ payload-aware guidance (ours)	85.0	100.0	94.1	62.5
	ACT	25.0	0.0	0.0	0.0
	Diffusion Policy	0.0	0.0	0.0	0.0

TABLE III
OUT-OF-DISTRIBUTION (OOD) ROBUSTNESS (%). SUCCESS RATES FOR VARIATIONS IN OBJECT CLASS AND GATE POSITIONING. *Place* AND *Hover* ARE REPORTED AS CONDITIONAL PERCENTAGES.

Grasp (Object Variation)			**Navigation (Gate Locations)**		
Object	**Pick**	**Place**	**Location**	**Gate**	**Hover**
Chips	10.0	0.0	Front	0.0	–
Sandwich	70.0	57.1	Left	0.0	–
Box	30.0	33.3	Right	40.0	100.0

inference as well as the necessity of GS synthetic trajectories to ensure successful gate traversal.

Out-of-distribution trials demonstrate generalization capabilities. Our method generalizes effectively in pick-and-place tasks, achieving up to 57% task success on a novel sandwich object. However, performance varies by object geometry: the model achieved only a 10% pick success rate with a bag of chips, compared to 70% for the sandwich. The primary failure mode was the policy's inability to adapt grasp poses to the chip bags geometry. In the navigation task, the policy adapts best to the "right" region (40% success). Conversely, trials in the "front" and "left" regions failed completely. Specifically, the drone consistently bypassed the gate in front-region trials, while left-region trials resulted in collisions due to incomplete position adaptation.

V. LIMITATIONS AND FUTURE WORK

We present AirVLA, the first systematic evaluation of transferring a manipulation-pretrained VLA foundation model to an aerial platform. Our results provide a nuanced answer to the challenge of cross-embodiment transfer: while the visual and semantic representations of models like π_0 transfer robustly to drone viewpoints, the intricate dynamics of aerial manipulation require specific interventions. By integrating payload-aware guidance directly into the flow-matching sampling loop, we improve single-task placement success from 0% (naive baseline) to 50%, reconciling the generative diversity of a VLA with the strict physical constraints of flight. Furthermore, we show that the policy can execute compositional tasks in a zero-shot manner, achieving 85% gate traversal and up to 62.5% place success rates. Although our method currently relies on motion capture, future work will target the use of on-board visual-inertial odometry (VIO) to replace this dependency. Finally, our out-of-distribution evaluations highlight a disparity in generalization: while the model achieves 57% success on novel object classes, the lower 40% success rate in novel navigation scenarios indicates that significantly more data is needed to enable robust aerial navigation.

REFERENCES

[1] Maximilian Adang, JunEn Low, Ola Shorinwa, and Mac Schwager. Singer: An onboard generalist vision-language navigation policy for drones. *arXiv preprint arXiv:2509.18610*, 2025.

[2] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[3] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, and et al. π_0: A vision-language-action flow model for general robot control, Nov 2024. URL https://doi.org/10.48550/arXiv.2410.24164.

[4] Kevin Black, Manuel Y Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. *arXiv preprint arXiv:2506.07339*, 2025.

[5] Karen Bodie, Maximilian Brunner, Michael Pantic, Stefan Walser, Patrick Pfändler, Ueli Angst, Roland Siegwart, and Juan Nieto. An omnidirectional aerial manipulation platform for contact-based inspection. *arXiv preprint arXiv:1905.03502*, 2019.

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina

Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *arXiv preprint arXiv:2212.06817*, 2022.

[7] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alex Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023.

[8] Qianzhong Chen, Naixiang Gao, Suning Huang, JunEn Low, Timothy Chen, Jiankai Sun, and Mac Schwager. Grad-nav++: Vision-language model enabled visual drone navigation with gaussian radiance fields and differentiable dynamics. *arXiv preprint arXiv:2506.14009*, 2025.

[9] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[10] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[11] Open X-Embodiment Collaboration et al. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

[12] Eugenio Cuniato, Ismail Geles, Weixuan Zhang, Olov Andersson, Marco Tognon, and Roland Siegwart. Learning to open doors with an aerial manipulator. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6942–6948. IEEE, 2023.

[13] Caiwu Ding, Lu Lu, Cong Wang, and Caiwen Ding. Design, sensing, and control of a novel uav platform for aerial drilling and screwing. *IEEE Robotics and Automation Letters*, 6(2):3176–3183, 2021.

[14] Vaibhav Ghadiok, Jeremy Goldin, and Wei Ren. Autonomous indoor aerial gripping using a quadrotor. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4645–4651. IEEE, 2011.

[15] Muhammad Fadhil Ginting, Sung-Kyun Kim, David D Fan, Matteo Palieri, Mykel J Kochenderfer, and Ali-akbar Agha-Mohammadi. Seek: Semantic reasoning for object goal navigation in real world inspection tasks. *arXiv preprint arXiv:2405.09822*, 2024.

[16] Xiaofeng Guo, Guanqi He, Jiahe Xu, Mohammadreza Mousaei, Junyi Geng, Sebastian Scherer, and Guanya Shi. Flying calligrapher: Contact-aware motion and force planning and control for aerial manipulation. *IEEE Robotics and Automation Letters*, 2024.

[17] Harsh Gupta, Xiaofeng Guo, Huy Ha, Chuer Pan, Muqing Cao, Dongjae Lee, Sebastian Sherer, Shuran Song, and Guanya Shi. Umi-on-air: Embodiment-aware guidance for embodiment-agnostic visuomotor policies. *arXiv preprint arXiv:2510.02614*, 2025.

[18] Huy Ha, Yihuai Gao, Zipeng Fu, Jie Tan, and Shuran Song. Umi on legs: Making manipulation policies mobile with manipulation-centric whole-body controllers. *arXiv preprint arXiv:2407.10353*, 2024.

[19] Guanqi He, Yash Janjir, Junyi Geng, Mohammadreza Mousaei, Dongwei Bai, and Sebastian Scherer. Image-based visual servo control for aerial manipulation using a fully-actuated uav. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023. URL https://arxiv.org/pdf/2306.16530.pdf.

[20] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

[21] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. *arXiv preprint arXiv:2210.05714*, 2022.

[22] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[23] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[24] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[25] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[26] Maximilian Laiacker, Felix Huber, and Konstantin Kon-

dak. High accuracy visual servoing for aerial manipulation using a 7 degrees of freedom industrial manipulator. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1631–1636, 2016. doi: 10.1109/IROS.2016.7759263.

[27] Christian Lanegger, Marco Ruggia, Marco Tognon, Lionel Ott, and Roland 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*, page p073, 2022.

[28] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

[29] JunEn Low, Maximilian Adang, Javier Yu, Keiko Nagami, and Mac Schwager. Sous vide: Cooking visual drone navigation policies in a gaussian splatting vacuum. *IEEE Robotics and Automation Letters*, 10(5): 5122–5129, 2025. doi: 10.1109/LRA.2025.3553785.

[30] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7086–7096, 2022.

[31] Keiko Nagami, Timothy Chen, Javier Yu, Ola Shorinwa, Maximilian Adang, Carlyn Dougherty, Eric Cristofalo, and Mac Schwager. Vista: Open-vocabulary, task-relevant robot exploration with online semantic gaussian splatting. *arXiv preprint arXiv:2507.01125*, 2025.

[32] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[33] Michael O'Connell, Guanya Shi, Xichen Shi, Kamyar Azizzadenesheli, Anima Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural-fly enables rapid learning for agile flight in strong winds. *Science Robotics*, 7(66): eabm6597, 2022.

[34] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

[35] Pedro Sanchez-Cuevas, Guillermo Heredia, and Anibal Ollero. Characterization of the aerodynamic ground effect and its influence in multirotor control. *International Journal of Aerospace Engineering*, 2017(1): 1823056, 2017.

[36] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023.

[37] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 conference proceedings*, pages 1–12, 2023.

[38] Justin Thomas, Giuseppe Loianno, Koushil Sreenath, and Vijay Kumar. Toward image based visual servoing for aerial grasping and perching. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2113–2118, 2014. doi: 10.1109/ICRA.2014.6907149.

[39] Miguel Angel Trujillo, José Ramiro Martínez-de Dios, Carlos Martín, Antidio Viguria, and Aníbal Ollero. Novel aerial manipulator for accurate and robust industrial ndt contact inspection: A new tool for the oil and gas inspection industry. *Sensors*, 19(6):1305, 2019.

[40] Anurag Sai Vempati, Mina Kamel, Nikola Stilinovic, Qixuan Zhang, Dorothea Reusser, Inkyu Sa, Juan Nieto, Roland Siegwart, and Paul Beardsley. Paintcopter: An autonomous uav for spray painting on three-dimensional surfaces. *IEEE Robotics and Automation Letters*, 3(4): 2862–2869, 2018. doi: 10.1109/LRA.2018.2846278.

[41] Robin Verschueren, Gianluca Frison, Dimitris Kouzoupis, Jonathan Frey, Niels van Duijkeren, Andrea Zanelli, Branimir Novoselnik, Thivaharan Albin, Rien Quirynen, and Moritz Diehl. acados—a modular open-source framework for fast embedded optimal control. *Mathematical Programming Computation*, 14 (1):147–183, 2022.

[42] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, et al. gsplat: An open-source library for gaussian splatting. *Journal of Machine Learning Research*, 26(34):1–17, 2025.

[43] Haijie Zhang, Jiefeng Sun, and Jianguo Zhao. Compliant bistable gripper for aerial perching and grasping. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1248–1253. IEEE, 2019.

[44] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023. URL https://arxiv.org/abs/2304.13705.

[45] Claudio Zito and Eliseo Ferrante. One-shot learning for autonomous aerial manipulation. *Frontiers in Robotics and AI*, 9:960571, 2022. doi: 10.3389/frobt.2022.960571. URL https://doi.org/10.3389/frobt.2022.960571.