Integrating Agglomerative Perception with One-step Action Generation for Robotic Manipulation

Sen Wang, *Student Member, IEEE,* Le Wang, *Senior Member, IEEE,* Sanping Zhou, *Member, IEEE,* Kun Xia, *Member, IEEE,* and Gang Hua, *Fellow, IEEE*

***Abstract*—Developing generalizable robotic policies that balance inference efficiency, manipulation accuracy, and robustness remains a formidable challenge. Existing vision-language-action models demand prohibitive data scales, while keyframe-based approaches struggle to reconcile the expressivity of generative models with the latency of iterative sampling. To address this trilemma, we present Flow2Act, a unified framework that integrates agglomerative perception with a deterministic one-step generative policy. Unlike prior methods relying on separate semantic encoders or iterative diffusion processes, our approach introduces three key innovations. First, we employ an agglomerative multi-teacher visual backbone that distills complementary strengths from diverse foundation models, capturing semantics, spatial structure, and segmentation to yield robust representations without task-specific pretraining. Second, we propose a conditional MeanFlow policy that parameterizes the interval-averaged velocity field. This formulation enables genuine single-step action generation, eliminating the discretization errors and computational overhead inherent in ODE-based flow matching. Third, we devise a curriculum region-aware mechanism via a Spatial-Grounded State Space Model, which progressively shifts attention from global flow stability to fine-grained contact precision. We evaluate Flow2Act on challenging simulation benchmarks and real-world robotic tasks, demonstrating significant gains in policy performance, robustness to environmental perturbations, and cross-task real-world applicability. Videos, code and more details are available at project page.**

***Index Terms*—Generalizable Robot Manipulation, Average Velocity Field, Region-Aware Perception.**

I. Introduction

ROBOTIC manipulation demands agents that generalize across objects and scenes while generating low-latency and accurate actions under realistic visual variability [1]–[5]. Developing such agents remains challenging, as it requires reasoning over multimodal observations, including visual perception, 3D spatial structure, and language instructions [6], [7]. These must be integrated into high-dimensional, physically plausible control outputs, often under limited supervision and in the presence of environmental variability.

Language-conditioned visuomotor policy learning has two dominant paradigms. The first paradigm is the end-to-end vision-language-action (VLA) model [8]–[14], which directly maps raw observations and instructions to low-level robot actions via a unified network. These VLA models are conceptually simple and demonstrate strong adaptability across diverse tasks and modalities by jointly learning the entire perception-to-action mapping. However, their primary limitation lies in the heavy reliance on large-scale training data, which typically requires tens of thousands of human demonstrations or robot interactions across hundreds of tasks [15], [16]. Collecting such large-scale datasets is costly and time-consuming, posing a significant barrier to real-world deployment.

The second paradigm adopts a keyframe-based strategy, predicting high-level end-effector poses for execution via a motion planner [17]. This strategy drastically reduces data requirements, while enabling complex manipulation behaviors [18]–[20]. Despite this progress, existing keyframe methods face a persistent, critical limitation: *Robotic policies struggle to balance generalization, efficiency, and accuracy within a single unified framework, as improving one aspect often compromises the others.* In response, we propose Flow2Act, a unified framework to tackle these trade-offs.

Generalization to Novel Scenes. A prevalent strategy for enhancing model generalization involves scaling training data. Recent research has expanded demonstration datasets through simulation, domain randomization, or large scale pretraining across diverse environments, yielding improved in distribution robustness [9], [21]–[24]. Nevertheless, robot data collection remains inherently time consuming and expensive. Even with extensive augmentation strategies, comprehensively covering the long tail distribution of variations in geometry, materials, lighting conditions, and occlusion patterns remains practically infeasible. An alternative approach leverages representation learning, utilizing pretrained visual models to extract transferable features before adapting them to manipulation tasks [19], [25], [26]. However, conventional 2D pretraining objectives, such as contrastive learning and masked image modeling, often lack sensitivity to contact scale geometry and occlusion dynamics critical for robotic manipulation. Guided by these

Manuscript received xxx; revised xxx; accepted xxx. Date of publication xxx; date of current version xxx. This work was supported in part by National Science and Technology Major Project under Grant 2024YFB4780100, National Natural Science Foundation of China under Grants 62088102 and U24A20325, Key Research and Development Plan of Shaanxi Province under Grant 2024PT-ZCK-80, and Fundamental Research Funds for the Central Universities under Grant XTR072022001. (*Corresponding author: Le Wang.*)

Sen Wang, Le Wang, and Sanping Zhou are with the National Key Laboratory of Human-Machine Hybrid Augmented Intelligence, National Engineering Research Center for Visual Information and Applications, and Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: sen.wang@stu.xjtu.edu.cn; lewang@mail.xjtu.edu.cn; spzhou@mail.xjtu.edu.cn;). Kun Xia is also with the School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: kunxia@mail.xjtu.edu.cn). Gang Hua is with Amazon Alexa AI, Bellevue, WA 98004 USA (e-mail: ganghua@gmail.com).

The source code is available at https://github.com/SanMumumu/Flow2Act

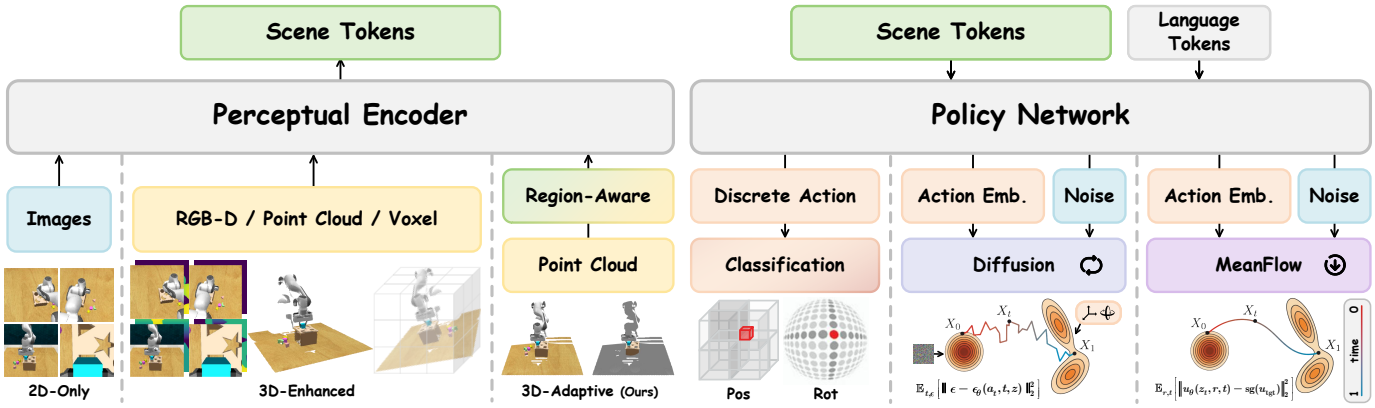

Fig. 1. **Architectural comparison of perceptual encoders and policy networks.** Left: 2D-only perception lacks critical 3D reasoning for manipulation, while 3D-enhanced methods (*e.g.*, RGB-D, point cloud and voxel) are constrained to global scene understanding; our region-aware schedule enables fine-grained, adaptive spatial perception. Right: Discrete action policies lack precision for manipulation, while diffusion and flow-based methods both learn invertible transformations to map prior Gaussian noise samples into the target action distribution. Diffusion-based approaches require iterative denoising steps to gradually refine noise into valid actions, whereas our MeanFlow framework directly learns a single-step mapping from noise to continuous actions. This unified policy network achieves higher computational efficiency and improved manipulation accuracy.

observations, our methodology avoids additional task specific pretraining of the visual backbone. Instead, we adopt a multi teacher agglomerative vision architecture as a unified encoder, effectively distilling complementary strengths from foundation models in semantic understanding, spatial reasoning, and scene segmentation [27]. This integration yields visual features exhibiting enhanced robustness under distribution shift while preserving contact level spatial detail. The architecture maintains computational efficiency during inference, providing a stronger foundation for our keyframe policy representation and one step action generation framework.

From Iterative Sampling to One-step Action Generation. A fundamental challenge in robotic policy design lies in balancing expressiveness with computational efficiency. Explicit methods discretize the action space into grids or value maps [28]–[31], enabling fast one-step prediction but suffering from quantization errors and poor scalability. In contrast, implicit generative models learn continuous, observation conditioned action distributions. In particular, diffusion models have become prominent in robotics due to their ability to represent complex, multimodal action distributions through iterative denoising [32]–[38]. However, diffusion models are inherently slow, each action requires tens or even hundreds of iterative denoising steps, making inference orders of magnitude slower than an explicit one-step prediction [39], [40]. To overcome the inefficiency of diffusion-based policies, recent approaches have proposed flow-based policies that perform deterministic action generation by integrating an ordinary differential equation (ODE) guided by a learned velocity field that continuously transports the noise distribution to the target action distribution [41]–[46]. Concretely, flow-based policies generate keyframes by transporting noise along probability flow paths, where the learned vector field captures multimodal action distributions. However, they often rely on explicit consistency constraints or numerical ODE solvers, which introduce structural limitations and can accumulate discretization error—especially under coarse integration steps. In this work, we reparameterize policy dynamics using the average velocity field over a time interval, making consistency an inherent property of the true field and eliminating the need for explicit constraints. This enables a closed-form, single-step update: a single forward pass maps a noise input directly to the target action keyframe, preserving accuracy while avoiding numerical errors from iterative ODE integration.

Curriculum Region-Aware Learning. Implicit policies operate in a latent space only loosely anchored to perception; the mapping from scene evidence to precise, contact-scale motions must be inferred from data, prolonging training. Without an explicit mechanism to localize task-relevant regions, capacity is dispersed over the full scene and fine details (*e.g.*, peg–hole alignment) are under-emphasized [18], [34], [47]. To address this gap, we design a region-aware flow policy with a simple curriculum learning. The model first learns a stable global flow. Once the velocity field is reliable, region awareness is activated to concentrate capacity on likely contact neighborhoods. This global-to-local schedule allows the generative flow to surface task-relevant areas and to recondition inference on high-resolution evidence while preserving single-step generation. The result is a tighter alignment between perception and action at the point of contact, which improves orientation control and accuracy under clutter and occlusion without extra model size or inference cost.

To this end, we introduce Flow2Act, a keyframe policy that co-designs perception and action by coupling an agglomerative multi-teacher visual backbone [27] with a single-step generative keyframe model in continuous, spatially grounded in the scene feature field. This pairing retains the spatial grounding of explicit maps and the expressivity of implicit models while removing their main bottlenecks, namely exponential discretization and multi-step sampling. Flow2Act delivers fixed-budget, real-time inference and contact-level precision, and exhibits strong transfer to unseen objects and scenes. Extensive experiments on RLBench [48] and Colosseum [49] show that the proposed method improves performance in both in-distribution and out-of-distribution settings. The model demonstrates better generalization, superior sample efficiency,

and faster inference compared to prior works.

In summary, our work presents three contributions:

- We introduce Flow2Act, a novel framework that couples an agglomerative visual backbone with a one-step generative model in continuous SE(3), yielding spatially grounded action prediction without discretization or iterative sampling.
- We reparameterize pose generation via interval-based transport, learning an average velocity field over a finite time horizon. This formulation intrinsically enforces consistency, enabling one-step inference while retaining the expressivity of implicit generative models.
- We propose a curriculum region-aware scheduling that first learns a stable average velocity field and then concentrates capacity on likely contact neighborhoods, improving orientation control and robustness under clutter and occlusion.

This paper extends our previous conference paper [41], and the new major contributions include:

- The incorporation of a vision foundation model [27] in place of CLIP, leading to improved robustness in realistic manipulation benchmarks with severe domain shifts.
- The introduction of a one-step action generation mechanism that improves inference efficiency and eliminates the need for ODE-based integration.
- The replacement of the region-aware fusion backbone with a stronger Mamba-2 [50] module that improves sequence modeling and multimodal alignment.
- A curriculum-guided region-aware learning strategy that progressively shifts attention from global scene understanding to contact-critical regions, stabilizing training and enhancing fine-grained manipulation accuracy.
- Additional experiments on the Colosseum benchmark [49] and new ablation studies are conducted to validate the effectiveness of each component.

The rest of the paper is organized as follows. Section II briefly reviews related work on robotic manipulation. Subsequently, we present the technical details of the proposed method in Section III. The experimental results are presented in Section IV. Finally, we conclude this paper in Section V.

II. Related Work

A. Visual Representations for Robotic Manipulation

Visual representation quality fundamentally determines the generalization capability of robotic manipulation policies. Early visuomotor approaches employed end-to-end convolutional architectures that processed raw RGB inputs directly [51]–[53]. While effective in constrained settings, such representations exhibit high sensitivity to texture variations, illumination changes, and background clutter, resulting in poor generalization to novel objects and unseen environments. The emergence of large vision-language models, particularly CLIP [54], marked a significant advancement. Works [18], [20], [55] demonstrated that integrating CLIP's semantic features with policy learning enables zero-shot generalization to previously unobserved objects. This capability allows robots to execute open-vocabulary instructions by grounding language in visual percepts. Nevertheless, CLIP and its successors [54], [56] prioritize image-level semantic alignment at the expense of spatial precision. These models lack sensitivity to contact-scale geometry and fine-grained structural details, which are crucial for high-accuracy manipulation tasks involving object parts such as handles, edges, or apertures. Subsequent research addressed this limitation through two complementary pathways. Self-supervised models [57], [58] provide dense spatial correspondences and implicit 3D shape priors, while segmentation foundation models [59] deliver pixel-accurate object masks essential for grasp localization. Contemporary Vision-Language-Action model consequently adopts multi-encoder architectures that concurrently process features from semantic, structural, and segmentation models [60], [61]. This strategy captures complementary visual signals but introduces substantial computational overhead during inference, impeding real-time deployment on robotic platforms.

Our approach leverages an agglomerative vision foundation model [27], a unified framework distilled from multiple pre-trained teacher models, including CLIP for semantic understanding, DINO [57] for dense spatial features, and SAM [59] for precise segmentation, enabling high-quality, multi-aspect visual representations in a single, efficient forward pass.

B. Flow Matching for Policy Learning

Diffusion-based visuomotor policies model action distribution through iterative denoising and have achieved strong performance across manipulation benchmarks [62]–[66]. Diffusion Policy [34] established that action diffusion enables flexible multi-modal prediction from visual observations, inspiring several extensions that emphasize spatial reasoning, including DP3 [67], HDP [35], ChainedDiffuser [37], and 3D Diffuser Actor [47]. Despite their effectiveness, diffusion models inevitably require multiple denoising steps at inference, which introduces latency and limits their applicability for real-time closed-loop control.

Flow-based approaches provide a more efficient alternative by learning deterministic transport mappings from noise to actions. Flow Matching and Rectified Flow [65], [68]–[70] regress velocity fields associated with straight-line probability paths and enable simpler training objectives than diffusion. Their conditional variants have shown promising results in imitation and reinforcement learning [43], [46], [71]. In robotic control, methods such as AdaFlow [43], π_0 [9] and FlowRAM [41] reduce sampling steps by solving state-conditioned flows, yet still rely on numerical ODE integration or require explicit consistency constraints to approximate one-step sampling. Consequently, their efficiency and stability remain tied to discretization quality.

The recently proposed MeanFlow paradigm [72] advances this line of research by replacing instantaneous velocity regression with the learning of interval-averaged velocities, enabling deterministic generation in a single network evaluation without ODE solvers or architectural constraints. MeanFlow has demonstrated high sample quality while providing genuine one-step inference, making it particularly suitable for time-critical control. Building on this insight, we adopt a conditional MeanFlow objective to realize one-step keyframe action generation directly from rich visuomotor observations,

achieving real-time inference while preserving the expressivity of continuous generative modeling.

C. Coordinating Perception and Policy

A longstanding challenge in visuomotor learning is the tight coupling between perception and action generation. High-level semantic understanding is essential for identifying task-relevant objects and affordances, while fine-grained spatial cues are required for accurate contact interactions and pose alignment. Prior works often address these two dimensions separately, leading to mismatches between what the perception module extracts and what the policy actually requires for precise control [8], [11], [60]. Several perception-driven frameworks improve visuomotor policies by enhancing visual representations. Vision foundation models such as CLIP-based encoders, DINO-style features, and language-grounded visual backbones have been shown to provide strong semantic priors that improve generalization across unseen objects and scenes [18], [35]. Meanwhile, 3D-centric architectures, such as voxel-based systems [18], point-based transformers [20], [73], and Gaussian-based spatial models [74], provide the high spatial fidelity needed for precise placement, grasping, and tool use. However, enhanced perception alone is insufficient when the policy module exhibits temporal instability or generates trajectories sensitive to local uncertainties. On the policy side, generative visuomotor models such as diffusion policies [34], [67] and flow-based formulations [41], [43] attempt to integrate perception by conditioning visual features into the denoising or flow transport process. While effective, these methods often assume that perceptual features are stable throughout training. In practice, unstable vector fields in early training stages may amplify irrelevant visual cues, leading to inaccurate keyframe proposals or degraded convergence. Recent works on region-aware or coarse-to-fine perception strategies [41], [75], [76] show that spatial selectivity is essential for bridging semantic context and spatial precision, but they require careful synchronization with the underlying policy dynamics. These studies collectively highlight that perception and policy cannot be optimized in isolation: visual features must evolve in tandem with the generative dynamics of the action model, and spatial selectivity must align with the stability of the underlying flow or diffusion process.

In this work, we build upon these insights and develop a coordinated framework that couples a task-aware visual encoder with a MeanFlow-based policy and a curriculum region-aware mechanism, ensuring that perceptual refinement and policy stabilization progress in a mutually consistent manner.

III. Methodology

A. Preliminaries: Rectified Flow

Rectified Flow [32], [68] is a family of generative models that learn a continuous-time velocity field to transport a source distribution π_0 to a target distribution π_1. Formally, a latent variable $\boldsymbol{z}_t$ evolves according to an ordinary differential equation (ODE):

$$d\boldsymbol{z}_t = \boldsymbol{v}_\theta(\boldsymbol{z}_t, t)dt, \quad t \in [0, 1], \tag{1}$$

where $\boldsymbol{v}_\theta(\boldsymbol{z}_t, t)$ denotes the instantaneous velocity field parameterized by a neural network. Given paired samples $(\boldsymbol{x}_0, \boldsymbol{x}_1)$ with $\boldsymbol{x}_0 \sim \pi_0$ and $\boldsymbol{x}_1 \sim \pi_1$, the intermediate latent is typically constructed via linear interpolation, $\boldsymbol{z}_t = (1-t)\boldsymbol{x}_0 + t\boldsymbol{x}_1$, and the corresponding ground-truth instantaneous velocity is defined as:

$$\boldsymbol{v}(\boldsymbol{z}_t, t) = \frac{d\boldsymbol{z}_t}{dt} = \boldsymbol{x}_1 - \boldsymbol{x}_0. \tag{2}$$

The training objective minimizes the MSE between the predicted velocity field and the ground-truth velocity:

$$\arg\min_{\boldsymbol{v}_\theta} \mathbb{E}_{t,(\boldsymbol{x}_0,\boldsymbol{x}_1)} \left[\|\boldsymbol{v}_\theta(\boldsymbol{z}_t, t) - \boldsymbol{v}(\boldsymbol{z}_t, t)\|^2 \right], \tag{3}$$

where $t \sim \text{Uniform}([0, 1])$. This regression objective encourages the model to approximate the underlying instantaneous velocity field along rectified flow paths. Once trained, samples are generated using the Euler method [65] with step size $\Delta t = 1/N$, where N is the number of time discretization steps from $t = 0$ to $t = 1$:

$$\hat{\boldsymbol{z}}_{t+\Delta t} = \hat{\boldsymbol{z}}_t + \boldsymbol{v}_\theta(\hat{\boldsymbol{z}}_t, t)\Delta t. \tag{4}$$

With the optimized $\boldsymbol{v}_\theta$ serving as a velocity field that drives the flow along nearly straight paths, Rectified Flow connects two distributions efficiently, allowing for high-quality generation with few discretization steps. In summary, the Rectified Flow framework provides an elegant formulation that closely approximates optimal transport.

B. Problem Definition

We consider the problem of multi-task robotic control, where a robot interprets visual observations and natural language instructions to generate executable actions. At each discrete time step t, the robot receives a visual input $\boldsymbol{o}_t$ and an instruction l, and produces an action:

$$\boldsymbol{a}_t = \mathcal{F}(\boldsymbol{o}_t, l; \theta), \tag{5}$$

where $\mathcal{F}$ denotes the policy mapping multimodal inputs to executable actions. The goal is to learn a generalizable policy that can perform diverse manipulation tasks across different scenes and linguistic conditions. Following previous arts [18], [20], [47], the training data consist of expert demonstrations $\mathcal{D} = (\zeta_i, l_i)_{i=1}^{N_D}$, where each trajectory $\{\boldsymbol{o}_i, \boldsymbol{a}_i\}_{i=1}^{N_\zeta}$ is represented as a sequence of keyframes capturing critical intermediate poses of end-effector. Each keyframe action is formatted as:

$$\boldsymbol{a} = \{\boldsymbol{a}_{\text{pos}} \in \mathbb{R}^3, \boldsymbol{a}_{\text{rot}} \in \mathtt{SO(3)}, \boldsymbol{a}_{\text{open}} \in \{0, 1\}\}, \tag{6}$$

where $\boldsymbol{a}_{\text{pos}}$ and $\boldsymbol{a}_{\text{rot}}$ denote the position and rotation of the end-effector, and $\boldsymbol{a}_{\text{open}}$ indicates the binary gripper state. A continuous 6D rotation representation is adopted to avoid quaternion discontinuities [41], [47]. During execution, the policy operates iteratively: (1) predicting the next action $\boldsymbol{a}_t$ conditioned on $(\boldsymbol{o}_t, l)$; (2) executing the motion toward the target keyframe pose T_t using a sampling-based planner; and (3) updating the observation until task completion or reaching a maximum step $S_{\max}$. This formulation unifies imitation learning and keyframe-based control, enabling scalable policy learning across diverse manipulation tasks.

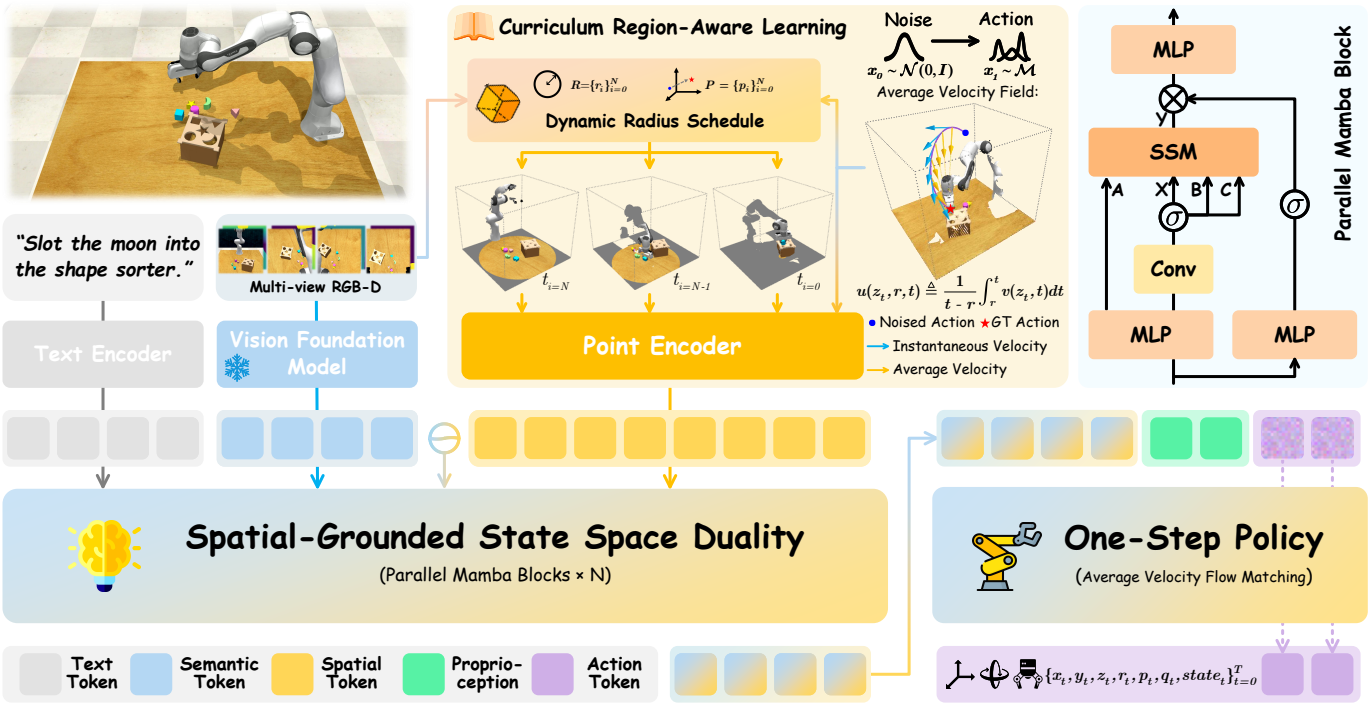

Fig. 2. **Overview of the Flow2Act framework.** The architecture unifies a multi-view perception module, a Spatial-Grounded State Space Duality, and a one-step policy head. Processing language instruction l and multiview RGB-D observations, the framework derives dense semantic encodings via an agglomerative vision foundation model alongside spatial tokens from a point encoder using curriculum region aware learning with a dynamic radius schedule. Stacked parallel Mamba blocks fuse these multimodal tokens to capture spatiotemporal dependencies. Subsequently, the one-step policy employs average velocity flow matching to synthesize precise robot actions $\boldsymbol{a}_t$ from initial noise ϵ_0.

C. Agglomerative Visual Perception

Robust robotic manipulation necessitates the synergistic integration of high-level semantic understanding and precise low-level spatial grounding. Formally, we denote the extracted semantic and spatial representations as $\boldsymbol{F}_{\text{sem}}$ and $\boldsymbol{F}_{\text{spat}}$, respectively. Existing frameworks typically adhere to a modality-decoupled paradigm, utilizing independent encoders, such as CLIP [54] for $\boldsymbol{F}_{\text{sem}}$ and point cloud encoder [77] for $\boldsymbol{F}_{\text{spat}}$. However, this separation often incurs substantial computational overhead and results in feature misalignment.

A pivotal challenge in generalizable robotic manipulation lies in the effective fusion of high-level semantic intent with low-level spatial constraints. Prevalent vision modules typically adhere to a modality-decoupled paradigm [60], [61], [78], employing an ensemble of specialized vision transformers processed in parallel, such as DINOv2 [57] for capturing dense spatial correspondences and SigLIP [56] for extracting rich semantic features. While effective in retrieving complementary information, this approach incurs substantial computational overhead due to the concurrent execution of multiple heavy backbones. Furthermore, it results in fragmented feature spaces that complicate downstream multimodal fusion.

To address these limitations, we integrate an agglomerative vision foundation model into our framework. Specifically, we leverage the RADIO architecture [27], which is engineered to distill the distinct capabilities of multiple pretrained teacher models, including CLIP [54], DINOv2, and SAM [59] into a unified student network. This unified formulation enables the extraction of state-of-the-art feature representations in a single forward pass, significantly enhancing inference efficiency. Crucially, the model inherits versatile capabilities from its teachers, such as zero-shot classification and open-set instance segmentation, with negligible architectural overhead [79], [80].

We empirically analyze the quality of these representations in Fig. 7. As illustrated by the PCA visualizations, standard semantic encoders like CLIP exhibit significant inconsistency across varying camera viewpoints, which is detrimental to spatial reasoning. Conversely, while DINOv3 [58] captures geometry, it often fails to cleanly separate task-relevant foregrounds from background clutter. In contrast, the employed agglomerative model demonstrates superior cross-view consistency and precise foreground grounding. This robustness against viewpoint shifts and background noise provides a critical perceptual foundation for precise manipulation in cluttered, unstructured environments.

Formally, within our Flow2Act framework, we instantiate the semantic and spatial representations as follows:

Semantic Features ($\boldsymbol{F}_{\text{sem}}$). We obtain $\boldsymbol{F}_{\text{sem}}$ by projecting the dense feature maps from the agglomerative backbone Φ_{agg}. Given input RGB images I, $\boldsymbol{F}_{\text{sem}}$ is defined as:

$$\boldsymbol{F}_{\text{sem}} = \Phi_{\text{agg}}(I), \tag{7}$$

where $\boldsymbol{F}_{\text{sem}}$ inherently encodes implicit 3D structural priors distilled from the teacher models providing a spatially consistent semantic anchor for the policy.

Spatial Features ($\boldsymbol{F}_{\text{spat}}$). Simultaneously, the spatial representation $\boldsymbol{F}_{\text{spat}}$ is derived from the point cloud P. Crucially, rather than employing a static encoding, the acquisition of

$\boldsymbol{F}_{\text{spat}}$ is modulated by a Dynamic Radius Schedule (detailed in Sec. III-F). This curriculum-driven mechanism adaptively regulates the perceptual scope of the point encoder [77], transitioning from global structural capture to fine-grained contact-aware features [20], [41], [76]. This ensures that $\boldsymbol{F}_{\text{spat}}$ encodes the most task-relevant spatial constraints adaptive to the policy's learning phase.

D. Spatial-Grounded State Space Duality

For robotic manipulation requiring high precision, merely concatenating multimodal features is insufficient; the policy must effectively reason about the *structural alignment* between the high-level semantic intent, the low-level geometric features, and the robot's physical state. We introduce the Spatial-Grounded State Space Duality (SG-SSD), a sequence modeling backbone designed to fuse these heterogeneous modalities while explicitly enforcing 3D spatial consistency.

Multimodal Sequence Construction. We first serialize the multi-source observations into a unified token sequence. The input sequence $\boldsymbol{F}_{\text{input}} \in \mathbb{R}^{L\times D}$ is constructed by concatenating the representations derived in Sec. III-C with other context features:

$$\boldsymbol{F}_{\text{input}} = \texttt{concat}(\boldsymbol{F}_{\text{sem}}, \boldsymbol{F}_{\text{geo}}, \boldsymbol{F}_{\text{text}}, \boldsymbol{F}_{\text{open}}), \tag{8}$$

where $\boldsymbol{F}_{\text{sem}}$ and $\boldsymbol{F}_{\text{geo}}$ are the agglomerative semantic features and adaptive geometric features, respectively. Crucially, each geometric token $\boldsymbol{f}_{\text{geo},i}$ is intrinsically linked to a physical 3D coordinate $\boldsymbol{p}_i \in \mathbb{R}^3$ in the robot's workspace. $\boldsymbol{F}_{\text{text}}$ and $\boldsymbol{F}_{\text{open}}$ denote linguistic and noise perturbed pose embeddings.

The Role of Spatial-Grounded Dynamics. Standard State Space Models (SSMs) [81], particularly the recent Mamba-2 architecture, introduce the Structured State Space Duality (SSD) framework [50]. This paradigm enables efficient training via matrix transformations while retaining linear recurrent inference. However, treating the input as a purely temporal signal inherently discards the critical 3D spatial relationships defining the manipulation environment. The objective of our Spatial-Grounded design is to explicitly inject geometric structure into the state dynamics without breaking the hardware-efficient attributes of the SSD algorithm. We leverage the parallel block design of Mamba-2, where parameters are projected at the start of the block. We modulate the input stream $\boldsymbol{x}_t$ by introducing a learnable spatial positional operator $\Phi(\boldsymbol{p}_i)$. The discrete state update equation is reformulated as:

$$\boldsymbol{h}_t = \boldsymbol{A}_t \boldsymbol{h}_{t-1} + \boldsymbol{B}_t \left(\boldsymbol{x}_t + \lambda \cdot \Phi(\boldsymbol{p}_i)\right), \tag{9}$$

where $\boldsymbol{A}_t$ and $\boldsymbol{B}_t$ denote the discretized state transition and input parameters, respectively. $\boldsymbol{x}_t$ is the semantic feature of the t-th token from $\boldsymbol{F}_{\text{input}}$, and λ is a learnable scalar that adaptively weighs the influence of the spatial prior. This formulation forces the latent state $\boldsymbol{h}_t$ to evolve based on both sequence content and the physical location of the geometric feature. Even when the sequence is arbitrarily ordered, this Spatial-Grounded Injection ensures that the model's dynamics maintain awareness of the geometric context, thereby achieving robust cross-modal alignment between the semantic, textual, and geometric inputs. Crucially, by injecting spatial information directly into the input token $\boldsymbol{x}_t$ rather than altering the recursive structure of $\boldsymbol{A}_t$, our formulation remains mathematically compatible with the efficient SSD matrix multiplication algorithm during training.

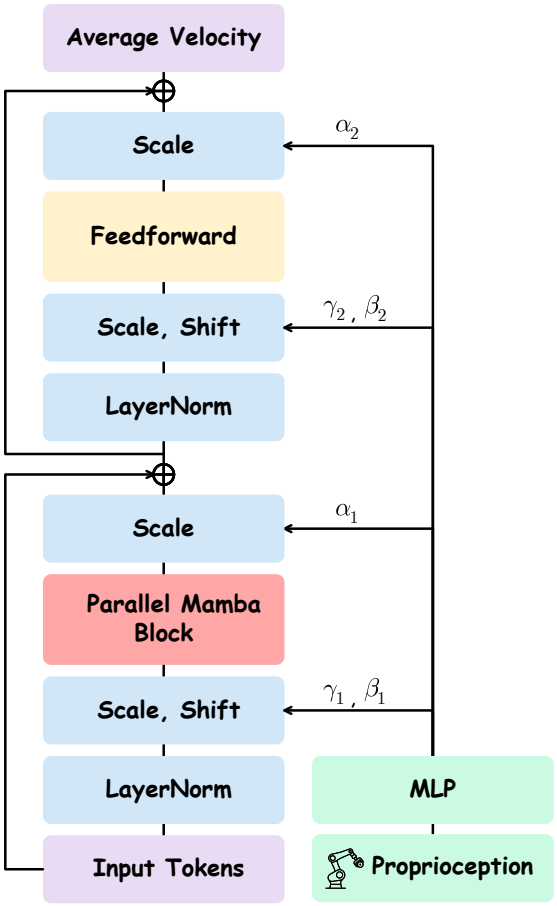

Fig. 3. **The Architecture of One-Step Policy Head.**

Adaptive Cross-Modal Gating. To further enhance fusion quality, we adopt the gated architecture inherent to Mamba-2 blocks. We leverage the parallel gating branch to dynamically control information flow:

$$\boldsymbol{F}_{\text{out}} = \text{Norm}\left(\text{SiLU}(\boldsymbol{F}_{\text{input}}\boldsymbol{W}_z) \odot \boldsymbol{y}_t\right), \tag{10}$$

where $\boldsymbol{W}_z$ projects the input into the gating subspace, SiLU is the activation function, and $\boldsymbol{y}_t$ is the output of the spatial-grounded SSD described in Eq. 9. The final Norm layer (*e.g.*, RMSNorm) ensures training stability. This mechanism allows the model to suppress noise from non-relevant modalities while amplifying task-critical geometric details.

E. One-step Policy Learning

Policy Formulation as Conditional MeanFlow. In the context of robotic policy learning, the objective is to learn a conditional mapping from multimodal observations to future actions. We formulate the policy as a conditional MeanFlow model, which parameterizes an average velocity field to deterministically transport a simple prior action distribution toward the expert action distribution. Let $\boldsymbol{a}_1 \sim \pi_1$ denote expert

Algorithm 1 MeanFlow Policy Training in Euclidean Space

Require: $\mathcal{D} = \{(\boldsymbol{o}, l, \boldsymbol{a})\}$: Expert demonstrations
repeat
  $\boldsymbol{a_0}, \boldsymbol{o}, l \sim \mathcal{D}$ # *Sample expert action (Target at $t = 0$)*
  $\boldsymbol{a_1} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ # *Sample noise (Prior at $t = 1$)*
  $r, t \sim \text{Uniform}[0, 1]$ with $r < t$ # *Sample time interval*
  $\boldsymbol{z_t} \leftarrow (1-t) \cdot \boldsymbol{a_0} + t \cdot \boldsymbol{a_1}$ # *Interpolate to state at time t*
  $\boldsymbol{v} \leftarrow \boldsymbol{a_1} - \boldsymbol{a_0}$ # *Instantaneous velocity*
  # *Compute total time derivative $\frac{d}{dt}u$ via JVP*
  $(\boldsymbol{u}, \dot{\boldsymbol{u}}) \leftarrow \texttt{jvp}(\boldsymbol{u}_\theta, (\boldsymbol{z_t}, r, t), (\boldsymbol{v}, 0, 1))$
  $\boldsymbol{u}_{\text{tgt}} \leftarrow \boldsymbol{v} - (t-r) \cdot \dot{\boldsymbol{u}}$ # *Apply MeanFlow Identity*
  $\mathcal{L}_{\text{MF}} \leftarrow \|\boldsymbol{u} - \texttt{sg}(\boldsymbol{u}_{\text{tgt}})\|^2$ # *MSE Loss with Stop-Gradient*
  $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{MF}}$
until Converged

Algorithm 2 One-step Action Generation

Require: $\boldsymbol{o}$: observation, l: instruction
  $\boldsymbol{a_1} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ # *Sample from Gaussian Prior (at $t = 1$)*
  $t \leftarrow 1$, $r \leftarrow 0$ # *From Noise(1) to Action(0)*
  $\hat{\boldsymbol{u}} \leftarrow \boldsymbol{u}_\theta(\boldsymbol{a_1}, r, t, \boldsymbol{o}, l)$ # *Predict average velocity over interval*
  $\boldsymbol{a_0} \leftarrow \boldsymbol{a_1} - (t-r) \cdot \hat{\boldsymbol{u}}$ # *One-step Backward Update*
  return $\boldsymbol{a_0}$ # *Return predicted action*

actions and $\boldsymbol{a}_0 \sim \pi_0 = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ denote actions drawn from a standard Gaussian prior. A latent action $\boldsymbol{z}_t$ is constructed via linear interpolation: $\boldsymbol{z}_t = (1-t)\boldsymbol{a}_0 + t\boldsymbol{a}_1$ where $t \in [0, 1]$. This defines a continuous trajectory in the action space $\mathbb{R}^9$, describing the transformation from the noisy prior to the expert action distribution. The neural architecture implementing this conditional mapping is illustrated in Fig. 3.

Average Velocity Field Modeling. Unlike conventional Flow Matching which models instantaneous velocity, MeanFlow [72] learns the average velocity field $\boldsymbol{u}_\theta(\boldsymbol{z}_t, r, t, |\boldsymbol{c})$. The ground-truth average velocity $\boldsymbol{u}$ is defined as:

$$\boldsymbol{u}(\boldsymbol{z}_t, r, t) = \frac{1}{t-r} \int_r^t \boldsymbol{v}(\boldsymbol{z}_\tau, \tau) d\tau, \tag{11}$$

where $\boldsymbol{v}$ represents the instantaneous velocity field. Differentiating over time yields the local relationship between them:

$$\boldsymbol{u}(\boldsymbol{z}_t, r, t) = \boldsymbol{v}(\boldsymbol{z}_t, t) - (t-r) \frac{d}{dt} \boldsymbol{u}(\boldsymbol{z}_t, r, t). \tag{12}$$

This identity relates the average velocity to the instantaneous velocity and its total time derivative, providing a principled supervision signal without requiring numerical integration during training.

During training, the network $\boldsymbol{u}_\theta(\boldsymbol{z}_t, r, t \mid \boldsymbol{c})$ is optimized to satisfy Eq. 12. The loss function is defined as:

$$\mathcal{L}_{\text{MF}}(\theta) = \mathbb{E}_{t, r, \boldsymbol{a}_0, \boldsymbol{a}_1} \left[\|\boldsymbol{u}_\theta(\boldsymbol{z}_t, r, t \mid \boldsymbol{c}) - \text{sg}(\boldsymbol{u}_{\text{tgt}})\|_2^2 \right], \tag{13}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator, which is essential to avoid optimizing through the target's Jacobian. The target velocity $\boldsymbol{u}_{\text{tgt}}$ is computed as:

$$\boldsymbol{u}_{\text{tgt}} = \boldsymbol{v}_t - (t-r) \left(\boldsymbol{v}_t \cdot \nabla_{\boldsymbol{z}} \boldsymbol{u}_\theta + \partial_t \boldsymbol{u}_\theta \right), \tag{14}$$

where $\boldsymbol{v}_t = \boldsymbol{a}_1 - \boldsymbol{a}_0$, the term $(\boldsymbol{v}_t \cdot \nabla_{\boldsymbol{z}} \boldsymbol{u}_\theta + \partial_t \boldsymbol{u}_\theta)$ represents the total time derivative $\frac{d}{dt}\boldsymbol{u}$, efficiently computed via Jacobian-Vector Products (JVP). The complete training procedure is formalized in Algorithm 1.

Simultaneously, to govern the prediction of the discrete end-effector state, we employ a binary cross-entropy objective, parameterized by a separate MLP head:

$$\mathcal{L}_{\text{open}}(\theta) = -\mathbb{E} \left[a_{\text{open}} \log(\hat{a}_{\text{open}}) + (1 - a_{\text{open}}) \log(1 - \hat{a}_{\text{open}}) \right], \tag{15}$$

where a_{open} and $\hat{a}_{\text{open}}$ denote the ground-truth and predicted probabilities of the gripper state, respectively. The total training objective is defined as the weighted sum of the meanflow loss $\mathcal{L}_{\text{MF}}(\theta)$ and the gripper loss $\mathcal{L}_{\text{open}}(\theta)$.

One-Step Deterministic Mapping. To generate an action in the inference phase, we sample $\boldsymbol{a}_0 \sim \pi_0$ from the prior and deterministically map it to the target expert action $\boldsymbol{a}_1$. By setting the interval from $r = 0$ to $t = 1$, the model performs a single-step update:

$$\boldsymbol{a}_1 = \boldsymbol{a}_0 + \boldsymbol{u}_\theta(\boldsymbol{a}_0, 0, 1 \mid \boldsymbol{c}). \tag{16}$$

This formulation enables the generation of high-fidelity keyframe actions in a single forward pass, strictly adhering to the physics of the learned average velocity field while eliminating the latency of iterative ODE solvers. The inference procedure is summarized in Algorithm 2.

F. Curriculum Region-Aware Learning

While the Dynamic Radius Schedule (DRS) proposed in FlowRAM [41] effectively balances global and local perception during inference, directly applying it from the onset of training introduces a critical instability. In the early training stages, the predicted velocity field $\boldsymbol{u}_\theta$ is randomly initialized, causing the noise-perturbed position $\boldsymbol{p}_i$ to drift far from the task-relevant region. Consequently, the region-aware encoder is forced to attend to irrelevant spatial noise, extracting misleading geometric features that further degrade policy optimization.

To mitigate this perception-action misalignment, we propose a Curriculum Region-Aware Learning strategy. Unlike FlowRAM, which couples region cropping with policy learning throughout the entire training process, we decouple this dependency into a coarse-to-fine schedule consisting of two distinct phases:

Phase I: Global Trajectory Stabilization (Coarse Stage). In the initial phase, we disable the region-aware mechanism. The perception module inputs the full-resolution global point cloud and semantic features extracted by the agglomerative backbone [27], utilizing a fixed, infinite radius $r \to \infty$. The objective in this stage is to stabilize the MeanFlow policy, encouraging the network to capture the global topology of the manipulation trajectory (*e.g.*, the reaching motion) without being distracted by fine-grained local geometric noise. This ensures that the predicted intermediate position $\boldsymbol{p}_i$ converges to a reliable neighborhood of the ground truth.

Phase II: Geometric Refinement (Fine Stage). Once the velocity field $\boldsymbol{u}_\theta$ stabilizes, we activate the Dynamic Radius Schedule. With a reliable trajectory prior, $\boldsymbol{p}_i$ now accurately

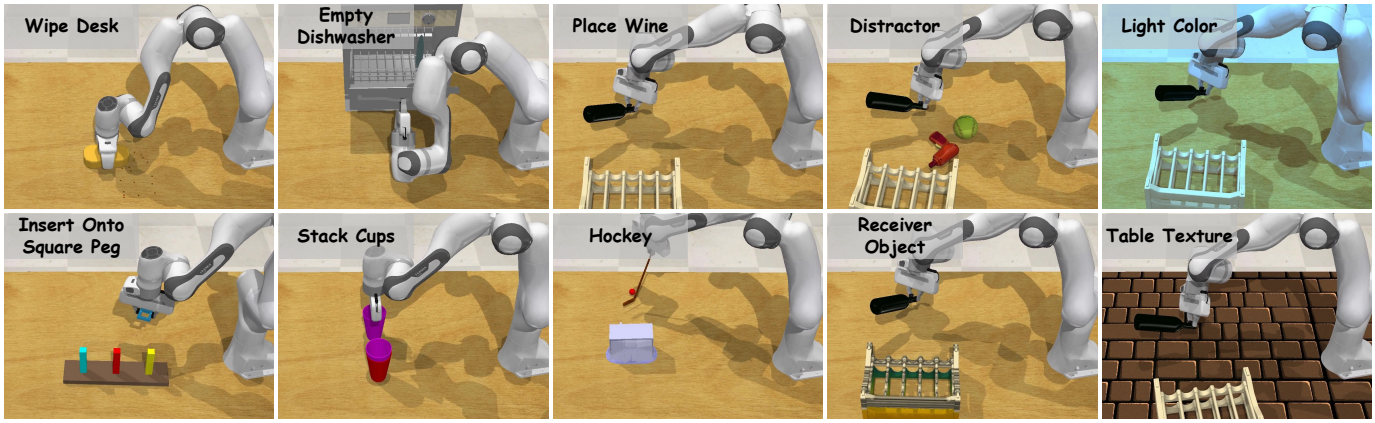

Fig. 4. **Overview of evaluated tasks.** The left three columns show a subset of standard RLBench tasks. The right two columns illustrate four distinct visual variants of the `Place Wine` task as defined in the COLOSSEUM benchmark.

indicates the region of interest. The model transitions to processing local geometric details using the variable radius r_i, defined as:

$$r_i = (1 - i) \cdot (r_0 - r_{\min}) + r_{\min}, \tag{17}$$

where i represents the normalized time step. This curriculum allows the policy to progressively shift its focus from global flow consistency to contact-rich geometric alignment, significantly enhancing manipulation precision.

This staged optimization strategy effectively breaks the vicious cycle of noisy perception and unstable control, leading to faster convergence and superior robustness compared to the original simultaneous training paradigm used in FlowRAM [41].

IV. Experiments and Analysis

We conduct comprehensive evaluations in both simulation and the real-world to assess the performance of the proposed method. These experiments are designed to investigate the following four questions systematically:

Q1: How effective is Flow2Act in learning robotic manipulation compared to state-of-the-art baselines?
Q2: How robust is Flow2Act to visual disturbances, such as distractors, backgrounds, and lighting variations?
Q3: How do different design choices contribute to the overall performance of Flow2Act?
Q4: How well does Flow2Act generalize to real-world robotic manipulation tasks?

A. Effectiveness on Robotic Manipulation

Environmental Setup. We conduct all simulation experiments on RLBench [48], a widely adopted robotic manipulation benchmark built upon the CoppelaSim simulator. We employ a 7-DoF Franka Emika Panda robot equipped with a parallel gripper to execute a diverse set of language-conditioned manipulation tasks. Visual observation configurations vary depending on the evaluation protocol. For multi-task [20], [73] and high-precision [41] evaluations, we utilize four calibrated RGB-D cameras positioned at the front, left shoulder, right shoulder, and wrist viewpoints. Conversely, for the few-shot setting [19], we use only the front-view camera to adhere to the GNFactor protocol. All input images are downsampled to a uniform resolution of 128×128. Figure 4 provides a visualization of the task.

We evaluate our method under three distinct experimental protocols: (1) Multi-Task setting: We adopt the standard 18-task subset from RLBench, a widely used benchmark for controlled multi-task evaluation. For each task, we collect 100 expert demonstrations using BiRRT-generated trajectories with hand-crafted waypoints. (2) High-Precision setting: To specifically evaluate fine-grained spatial understanding and precise manipulation capabilities, we select 7 high-precision tasks from RLBench that require accurate alignment of small objects (*e.g.*, `Insert USB`, `Screw Nail`, `Insert onto Square Peg`). We collect 100 expert demonstrations per task, as in the multi-task setting. (3) Few-Shot setting: We evaluate on 10 language-conditioned manipulation tasks from RLBench under extreme data limitations. Each task contains only 20 expert demonstrations. The dataset includes at least two variations in object attributes per task (shape, color, or spatial configuration), resulting in a total of 166 distinct task instances. This minimal-data setting provides a rigorous benchmark for evaluating generalization capabilities with limited supervision. For all three protocols, each policy is evaluated over 25 rollout episodes per task variant, and we report the average task success rate across all variants.

Comparisons and Baselines. All methods compared on RLBench utilize 3D information. We contrast Flow2Act with the previous state-of-the-art: C2F-ARM-BC [53] predicts the next keyframe action in the voxel space with a coarse-to-fine strategy. PerAct [18] also voxelizes the 3D workspace and employs a perceiver transformer. 3D-MVP [25] leverages 3D multiview masked autoencoding for pretraining, enhancing generalization via large-scale 3D-aware representation learning. Act3D [76] applies adaptive resolution 3D point sampling to generate hierarchical resolution 3D action maps. 3D Diffuser Actor (3DDA) [47] unifies diffusion policies and 3D scene representations, leveraging a 3D denoising transformer

TABLE I
EVALUATION RESULTS OF MULTI-TASK ON RLBENCH. EACH TASK IS EVALUATED WITH 25 ROLLOUTS UNDER 3 DIFFERENT SEEDS. WE REPORT THE AVERAGE SUCCESS RATE AND STANDARD DEVIATION FOR ALL TASKS. VARIANCES ARE INCLUDED WHEN AVAILABLE. THE "AVG. RANK" COLUMN REPORTS THE AVERAGE RANK OF EACH METHOD ACROSS ALL PERTURBATIONS, WHERE LOWER VALUES INDICATE BETTER OVERALL PERFORMANCE.

Method / Task	Avg. Success↑	Avg. Rank↓	Push Buttons	Slide Block	Sweep to Dustpan	Meat off Grill	Turn Tap	Put in Drawer	Close Jar	Drag Stick
C2F-ARM-BC [53]	20.1	8.7	72.0	16.0	0.0	20.0	68.0	4.0	24.0	24.0
PerAct [18]	49.4	6.3	92.8 ± 3.0	74.0 ± 13.0	52.0 ± 0.0	70.4 ± 2.0	88.0 ± 4.4	51.2 ± 4.7	55.2 ± 4.7	89.6 ± 4.1
3D-MVP [25]	67.5	3.5	100.0	48.0	80.0	96.0	96.0	100.0	76.0	100.0
Act3D [76]	65.0	4.3	99.0	93.0	92.0	94.0	94.0	90.0	92.0	92.0
3DDA [47]	81.3	2.5	98.4 ± 2.0	97.6 ± 3.2	84.0 ± 4.4	96.8 ± 1.6	99.2 ± 1.6	96.0 ± 3.6	96.0 ± 2.5	100.0 ± 0.0
RVT-2 [20]	81.4	2.5	100.0 ± 0.0	92.0 ± 2.8	100.0 ± 0.0	**99.0** ± 1.7	99.0 ± 1.7	**96.0** ± 0.0	**100.0** ± 0.0	99.0 ± 1.7
FlowRAM [41]	84.9	2.2	100.0 ± 0.0	100.0 ± 0.0	92.0 ± 2.0	94.0 ± 2.0	100.0 ± 0.0	92.0 ± 0.0	96.0 ± 2.0	100.0 ± 0.0
Flow2Act (ours)	**87.1**	**1.7**	**100.0** ± 0.0	**100.0** ± 0.0	**100.0** ± 2.0	97.7 ± 2.0	**100.0** ± 0.0	93.3 ± 2.7	98.0 ± 2.0	**100.0** ± 0.0

Method	Put in Safe	Place Wine	Screw Bulb	Open Drawer	Stack Blocks	Stack Cups	Put in Cupboard	Insert Peg	Sort Shape	Place Cups
C2F-ARM-BC [53]	12.0	8.0	8.0	20.0	0.0	0.0	0.0	4.0	8.0	0.0
PerAct [18]	86.0 ± 3.2	44.8 ± 7.8	17.6 ± 2.0	88.0 ± 5.7	26.4 ± 3.9	2.4 ± 2.2	28.0 ± 4.4	5.6 ± 4.1	16.8 ± 4.7	2.4 ± 3.2
3D-MVP [25]	92.0	100.0	60.0	84.0	40.0	36.0	60.0	20.0	28.0	4.0
Act3D [76]	95.0	80.0	47.0	93.0	12.0	9.0	51.0	27.0	8.0	3.0
3DDA [47]	97.6 ± 2.0	93.6 ± 4.8	82.4 ± 2.0	89.6 ± 4.1	68.3 ± 3.3	47.2 ± 8.5	85.6 ± 4.1	65.6 ± 4.1	44.0 ± 4.4	24.0 ± 7.6
RVT-2 [20]	96.0 ± 2.8	95.0 ± 3.3	**88.0** ± 4.9	74.0 ± 11.8	80.0 ± 2.8	**69.0** ± 5.9	66.0 ± 4.5	40.0 ± 0.0	35.0 ± 7.1	38.0 ± 4.5
FlowRAM [41]	96.0 ± 0.0	96.0 ± 0.0	84.0 ± 2.3	92.0 ± 0.0	77.3 ± 3.8	61.0 ± 2.0	86.0 ± 4.0	72.0 ± 2.7	48.0 ± 4.0	42.0 ± 2.3
Flow2Act (ours)	**98.0** ± 0.0	**97.7** ± 2.0	86.3 ± 2.7	**96.0** ± 4.0	**80.0** ± 2.0	64.0 ± 3.3	**88.0** ± 0.0	**74.7** ± 3.7	**51.3** ± 2.0	**44.0** ± 3.3

to predict action sequences. RVT-2 [20] employs a multi-stage inference pipeline and leverages virtual viewpoint rendering to capture detailed 3D scene information. FlowRAM [41] learns region-aware flow fields for manipulation by integrating spatial token grounding with temporal correspondence, enabling precise and robust keyframe prediction. GNFactor [19] jointly optimizes a generalizable neural field to enhance 3D understanding.

Implementation Details. We employ the AdamW [82] optimizer with a base learning rate of $1e^{-4}$, linear warmup over the first 5K steps, and cosine annealing scheduling. Our training protocol spans 300K total steps with a batch size of 320, applying EMA to model weights for stability. Following our curriculum region-aware strategy, the first 200K steps focus exclusively on learning the global mean velocity field, while the final 100K steps gradually integrate region-aware perception to refine contact-scale accuracy. For fair comparison against prior state-of-the-art methods [20], [41], all models, including reproduced baselines, are trained and evaluated on identical hardware configurations (8 NVIDIA RTX 3090 GPUs), with inference speed measurements performed on a single RTX 3090 using identical input data.

Results on Multi-Task setting. In the multi-task setting, which evaluates performance across 18 diverse RLBench tasks, Flow2Act demonstrates superior effectiveness compared to all state-of-the-art baselines. As shown in Table I, our method achieves a new state-of-the-art average success rate of **87.1**%, significantly outperforming previous top performers like RVT-2 [20] (81.4%) and FlowRAM [41] (84.9%). This high average success rate indicates that Flow2Act can learn robust policies capable of handling a wide variety of manipulation tasks. Furthermore, Flow2Act excels in specific challenging tasks. For instance, it achieves perfect success rates (100.0%) on tasks such as `Push Buttons`, `Slide Block`, `Sweep to Dustpan`, `Turn Tap`, and `Drag Stick`. Notably, it also shows strong performance on tasks requiring precise interaction, such as `Insert Peg` (74.7%) and `Sort Shape` (51.3%), where many prior methods struggle. The consistently high scores across nearly all 18 tasks highlight Flow2Act's ability to generalize effectively across different task categories, establishing it as a highly effective solution for multi-task robotic manipulation.

TABLE II
EVALUATION RESULTS ON HIGH-PRECISION TASK ON RLBENCH.

Method / Task	Avg. Success↑	Screw Nail	Insert onto Square Peg	Plug Charger
Act3D [76]	20.2	$28.0^{\pm 3.3}$	$6.7^{\pm 3.7}$	$15.3^{\pm 2.3}$
3DDA [47]	40.0	$48.0^{\pm 1.6}$	$45.9^{\pm 2.6}$	$30.7^{\pm 5.3}$
RVT-2 [20]	39.4	$50.7^{\pm 8.5}$	$53.1^{\pm 6.0}$	$34.7^{\pm 7.1}$
FlowRAM [41]	52.0	$54.7^{\pm 1.9}$	$69.3^{\pm 7.5}$	$52.0^{\pm 8.6}$
Flow2Act (ours)	**55.2**	**57.3**$^{\pm 1.9}$	**72.0**$^{\pm 3.7}$	**54.0**$^{\pm 3.3}$

Method / Task	Setup Checkers	Insert USB	Unplug Charger	Put umbrella in Stand
Act3D [76]	$37.3^{\pm 6.8}$	$10.3^{\pm 5.7}$	$37.3^{\pm 5.3}$	$6.7^{\pm 4.7}$
3DDA [47]	$46.7^{\pm 3.3}$	$47.7^{\pm 1.6}$	$44.7^{\pm 5.3}$	$16.0^{\pm 3.3}$
RVT-2 [20]	$62.7^{\pm 5.3}$	$21.3^{\pm 9.2}$	$45.3^{\pm 6.1}$	$8.0^{\pm 0.0}$
FlowRAM [41]	$66.7^{\pm 6.1}$	$57.3^{\pm 3.2}$	$46.7^{\pm 1.9}$	$17.3^{\pm 9.4}$
Flow2Act (ours)	**69.7**$^{\pm 4.7}$	**59.7**$^{\pm 4.7}$	**49.7**$^{\pm 2.7}$	**24.0**$^{\pm 4.0}$

Results on High-precision setting. For high-precision tasks, which demand fine-grained spatial understanding and accurate alignment, Flow2Act proves to be exceptionally ef-

TABLE III
EVALUATION RESULTS OF FEW-SHOT SETTING ON RLBENCH.

Method / Task	Avg. Success↑	Close Jar	Open Drawer	Sweep to Dustpan	Meat off Grill	Turn Tap	Slide Bolck	Put in Drawer	Drag Stick	Put Buttons	Stack Blocks
PerAct [18]	20.4	$18.7^{\pm 13.6}$	$54.7^{\pm 18.6}$	$0.0^{\pm 0.0}$	$40.0^{\pm 17.0}$	$38.7^{\pm 6.8}$	$18.7^{\pm 13.6}$	$2.7^{\pm 3.3}$	$5.3^{\pm 3.5}$	$18.7^{\pm 12.4}$	$6.7^{\pm 1.9}$
GNFactor [19]	31.7	$25.3^{\pm 6.8}$	$76.0^{\pm 5.7}$	$28.0^{\pm 15.0}$	$57.3^{\pm 18.9}$	$50.7^{\pm 8.2}$	$20.0^{\pm 15.0}$	$0.0^{\pm 0.0}$	$37.3^{\pm 13.2}$	$18.7^{\pm 10.0}$	$4.0^{\pm 3.3}$
Act3D [76]	65.3	$52.0^{\pm 5.7}$	$84.0^{\pm 8.6}$	$80.0^{\pm 9.8}$	$66.7^{\pm 1.9}$	$64.0^{\pm 5.7}$	**100.0**$^{\pm 0.0}$	$54.7^{\pm 3.8}$	$86.7^{\pm 1.9}$	$64.0^{\pm 1.9}$	$0.0^{\pm 0.0}$
3DDA [47]	78.4	$82.7^{\pm 1.9}$	$89.3^{\pm 7.5}$	**94.7**$^{\pm 1.9}$	**88.0**$^{\pm 5.7}$	$80.0^{\pm 8.6}$	$92.0^{\pm 0.0}$	$77.3^{\pm 3.8}$	$98.7^{\pm 1.9}$	$69.3^{\pm 5.0}$	$12.0^{\pm 3.7}$
RVT-2 [20]	76.2	$79.3^{\pm 5.3}$	$78.7^{\pm 2.7}$	$87.3^{\pm 6.7}$	$86.7^{\pm 6.7}$	$85.3^{\pm 2.9}$	$76.7^{\pm 9.5}$	$86.7^{\pm 6.7}$	$96.0^{\pm 0.0}$	$67.3^{\pm 1.1}$	$27.7^{\pm 4.0}$
FlowRAM [41]	82.3	$85.0^{\pm 3.7}$	$90.0^{\pm 3.3}$	$88.0^{\pm 5.6}$	$82.0^{\pm 1.9}$	$86.3^{\pm 7.7}$	$93.3^{\pm 1.9}$	$88.0^{\pm 5.6}$	$100.0^{\pm 0.0}$	$80.3^{\pm 3.7}$	$31.3^{\pm 2.8}$
Flow2Act (ours)	**85.0**	**88.0**$^{\pm 0.0}$	**93.3**$^{\pm 1.7}$	$90.7^{\pm 7.5}$	$86.0^{\pm 0.0}$	**94.7**$^{\pm 3.8}$	$89.3^{\pm 1.9}$	**90.0**$^{\pm 2.0}$	**100.0**$^{\pm 0.0}$	**82.3**$^{\pm 2.3}$	**36.0**$^{\pm 2.0}$

TABLE IV
EVALUATION RESULTS ON COLOSSEUM. THE TABLE SHOWS THE SUCCESS RATES ACROSS 14 GENERALIZATION SETTINGS. THE "AVG. RANK" COLUMN REPORTS THE AVERAGE RANK OF EACH METHOD ACROSS ALL PERTURBATIONS, WHERE LOWER VALUES INDICATE BETTER PERFORMANCE.

Method / Task	Avg. Success↑	Avg. Rank↓	All Perturbations	MO-COLOR	RO-COLOR	MO-TEXTURE	RO-TEXTURE	MO-SIZE
R3M-MLP [83]	0.8	5.71	0.6	0.4	0.0	0.0	0.0	1.8
MVP-MLP [84]	1.6	5.0	0.8	1.2	0.0	0.4	0.0	4.44
PerAct [18]	27.9	3.71	7.2	24.0	29.2	28.8	17.71	35.3
RVT-2 [20]	56.7	1.92	15.6 $^{\pm 0.8}$	53.0 $^{\pm 0.9}$	54.6 $^{\pm 0.6}$	**59.7** $^{\pm 0.7}$	56.7 $^{\pm 1.4}$	60.9 $^{\pm 0.9}$
FlowRAM [41]	55.2	1.78	13.5 $^{\pm 1.0}$	51.0 $^{\pm 1.0}$	55.3 $^{\pm 0.8}$	55.0 $^{\pm 0.9}$	53.0 $^{\pm 1.2}$	62.7 $^{\pm 1.0}$
Flow2Act (ours)	**58.4**	**1.51**	**17.5** $^{\pm 2.0}$	**54.3** $^{\pm 1.1}$	**56.7** $^{\pm 1.0}$	57.3 $^{\pm 2.0}$	**57.3** $^{\pm 1.0}$	**64.3** $^{\pm 1.0}$

Method / Task	RO-SIZE	Light Color	Table Color	Table Texture	Distractor	Background Texture	RLBench	Camera Pose
R3M-MLP [83]	0.0	1.0	1.4	0.2	1.6	1.2	2.0	0.8
MVP-MLP [84]	0.0	1.6	1.6	1.0	3.8	2.2	2.0	2.6
PerAct [18]	29.3	29.1	30.4	23.2	27.1	33.5	39.4	36.3
RVT-2 [20]	53.4 $^{\pm 1.5}$	58.0 $^{\pm 1.1}$	**62.6** $^{\pm 0.9}$	56.6 $^{\pm 0.9}$	60.8 $^{\pm 0.5}$	68.7 $^{\pm 1.1}$	68.8 $^{\pm 1.3}$	**64.4** $^{\pm 0.5}$
FlowRAM [41]	55.0 $^{\pm 1.3}$	61.3 $^{\pm 1.0}$	59.0 $^{\pm 0.8}$	53.0 $^{\pm 0.7}$	58.0 $^{\pm 0.6}$	70.0 $^{\pm 1.0}$	66.0 $^{\pm 1.2}$	60.0 $^{\pm 0.7}$
Flow2Act (ours)	**58.3** $^{\pm 1.7}$	**63.3** $^{\pm 1.2}$	61.7 $^{\pm 0.9}$	**58.0** $^{\pm 0.7}$	**62.0** $^{\pm 1.5}$	**74.3** $^{\pm 1.0}$	**70.3** $^{\pm 0.8}$	62.3 $^{\pm 0.8}$

fective, achieving the best performance among all evaluated methods. The results Table II demonstrate that Flow2Act achieves the highest average success rate of **55.2**% and outperforms all previous methods by a wide margin across all tasks. These results clearly show that Flow2Act's design is particularly well-suited for high-precision scenarios, where even minor errors can lead to failure. Its ability to consistently outperform other SOTA methods in these demanding tasks underscores its effectiveness in learning the intricate spatial reasoning required for precise robotic control.

Results on Few-Shot setting. In the few-shot setting, which evaluates generalization under extremely limited supervision (only 20 demonstrations per task), Flow2Act exhibits remarkable effectiveness, setting a new benchmark for data-efficient learning. As detailed in Table III, Flow2Act achieves an impressive average success rate of **85.0**%, significantly outperforming all previous methods. This result is particularly noteworthy because it demonstrates that Flow2Act can learn highly effective policies with minimal expert data. It not only maintains high performance on standard tasks but also excels in tasks that are typically difficult to learn from few examples. This improvement is not limited to easy tasks; for instance, on the challenging `Insert onto Square Peg` task, Flow2Act achieves 72.0% success, notably higher than both FlowRAM [41] (69.3%) and RVT-2 [20] (53.1%). The consistent outperformance across all 10 tasks in this low-data regime highlights Flow2Act's powerful generalization capabilities. It can extract meaningful patterns and build robust policies from very sparse demonstrations, making it a highly effective framework for real-world applications where collecting large amounts of expert data is often impractical.

These results collectively prove that Q1: Flow2Act is highly effective in learning robotic manipulation compared to state-of-the-art baselines, delivering superior performance, robustness, and data efficiency across diverse task complexities and data regimes.

B. Robustness to Visual Disturbances

Environmental Setup. To evaluate Flow2Act's robustness to visual disturbances such as distractor objects, background variations, and lighting changes, we conduct tests on the COLOSSEUM benchmark [49], an extension of RLBench designed to assess generalization under unseen visual conditions. The model is trained solely on clean RLBench data, with 100 demonstrations per task across 20 tasks and no exposure to visual perturbations. During evaluation, it is tested on 12 distinct perturbation types that were never seen during training, including altered object textures and colors, complex or changing backgrounds, variable illumination levels, the addition of distractor objects, and modified camera viewpoints. These perturbations collectively form 20,371 unique visual scenarios, enabling a comprehensive assessment of the model's

TABLE V
COMPARISON OF ARCHITECTURES AND MODULES. WE REPORT THE AVERAGE SUCCESS RATE (%) AND INFERENCE TIME (MS) ACROSS ALL EPISODES. "DISTURBANCES" REFER TO "ALL PERTURBATIONS" IN THE COLOSSEUM BENCHMARK.

Type	Ablation	Multi-Task	High-Precision	Few-Shot	Disturbances	Infer
Our Complete Architecture	**Flow2Act**	**87.1**	**55.2**	**85.0**	**17.5**	**35.3**
Agglomerative Visual Perception (Sec III-C)	w/o. Vision Foundation Model	85.4 (**-1.7**)	52.4 (-2.8)	82.7 (**-2.3**)	14.2 (**-3.3**)	30.7
Spatial-Grounded State Space Duality (Sec III-D)	w/o. State Space Duality	86.4 (-0.7)	54.0 (-1.2)	84.0 (-1.0)	17.1 (-0.4)	115.3
One-step Policy Learning (Sec III-E)	w/o. MeanFlow	85.8 (-1.3)	53.7 (-1.5)	83.2 (-1.8)	16.6 (-0.9)	865.7
Curriculum Region-Aware Learning (Sec III-F)	w/o. Course Learning	86.2 (-0.9)	53.1 (-2.1)	83.5 (-1.5)	17.1 (-0.4)	35.3
	w/o. Dynamic Radius Scheduling	85.6 (-1.5)	51.3 (**-3.9**)	82.9 (-2.1)	16.8 (-0.7)	32.7

ability to generalize under real-world visual variability. For each task–perturbation combination, we perform 25 independent trials and compute the mean success rate across all tasks for each disturbance type. We also report performance on the standard RLBench setting without perturbations (denoted as "RLBench" in Table IV) and under the most challenging condition where all 12 perturbations are applied simultaneously (denoted as "All Perturbations" in Table IV).

Comparisons and Baselines. Our evaluation includes five comparative approaches across different architectural paradigms. Two 2D-based techniques employ pre-trained vision backbones for observation processing: R3M-MLP [84] leverages R3M, a model trained on extensive egocentric video datasets, while MVP-MLP utilizes MVP [83], which has been pretrained on diverse real-world visual data. These vision encoders have demonstrated robust performance across multiple robotic scenarios in both simulated and physical environments. Additionally, we evaluate against three 3D-aware methods detailed in Section IV-A: PerAct [18], RVT-2 [20] and FlowRAM [41], which represent state-of-the-art approaches in spatially-aware robotic learning.

Results on COLOSSEUM. To evaluate the robustness of Flow2Act to visual disturbances, we conduct comprehensive experiments on the COLOSSEUM benchmark, which assesses generalization capabilities under 14 diverse environmental perturbations across 20 tasks. These perturbations include variations in object color (MO-COLOR, RO-COLOR), object texture (MO-TEXTURE, RO-TEXTURE), object size (MO-SIZE, RO-SIZE), lighting conditions, table color and texture, distractors, background texture, RLBench variations, and camera pose changes [49]. As show in Table IV, Flow2Act demonstrates superior robustness across all perturbation categories, achieving an average success rate of **58.4**% with an average rank of **1.51** across all 14 settings. This represents a significant improvement over the previous state-of-the-art method, FlowRAM (55.2% average success rate). Notably, Flow2Act excels in challenging scenarios involving distractors (62.0% success) and background texture variations (74.3% success), where even minor visual disturbances typically cause substantial performance degradation in other methods. The model also shows remarkable resilience to object size variations, achieving the highest success rate of 64.3% in the MO-SIZE category.

These results confirm that Flow2Act's representation learning framework effectively captures spatial and physical priors that enable robust performance under diverse visual disturbances, without requiring explicit domain randomization during training. The consistent outperformance across all perturbation types demonstrates Flow2Act's strong capability to generalize to unseen environmental variations, addressing Q2 affirmatively.

C. Ablation Study & Design Analysis

Analysis of Architecture and Module. To meticulously evaluate the contribution of each component within the Flow2Act framework, we conducted extensive ablation studies. Table V presents the quantitative results, which detail the success rates across varying benchmarks and the average inference latency. (1) Removing the vision foundation model results in a performance attenuation across all metrics. Notably, the success rate in the Disturbances setting drops significantly by 3.3% to 14.2%. This empirical evidence substantiates that distilling knowledge from multiple teacher models is essential for maintaining robustness against severe environmental variations and visual perturbations. (2) The ablation of the State Space Duality module leads to a marginal decrease in success rates but precipitates a dramatic increase in inference latency from 35.3 ms to 115.3 ms. This finding validates the architectural superiority of our design. It confirms that the Mamba-based backbone efficiently fuses heterogeneous modalities with linear complexity while avoiding the high computational overhead typical of traditional attention mechanisms. (3) Substituting the MeanFlow objective with standard flow matching methodologies results in a prohibitive increase in inference time to 865.7 ms. Furthermore, the Multi-Task success rate declines by 1.3%. This comparison demonstrates that our reparameterized average velocity field not only enables real-time responsiveness through genuine one-step generation but also preserves the expressivity required for high-fidelity action prediction. (4) The exclusion of the Dynamic Radius Scheduling mechanism causes the most severe performance degradation in the High-Precision setting with a 3.9% drop. This sharp decline proves that the coarse-to-fine perceptual schedule is vital for resolving contact-rich manipulation tasks. Additionally, removing the Course Learning strategy leads to a 2.1% drop in high-precision tasks, indicating that stabilizing the global trajectory before refining local details is crucial for optimal policy convergence.

Fig. 5. **Real-world hardware platforms and visualizations of sampled tasks.** We evaluate on six manipulation tasks: `Stack Blocks`, `Hanging Cup`, `Put Mustard`, `Stack Cups`, `Put Item in Drawer`, `Sort Shape`. To assess generalization, we introduce six types of disturbances in the basic setting, including changes in `Background`, `Distractors`, `Spatial Location`, `Object Appearance`, `Camera View`, `Lighting`.

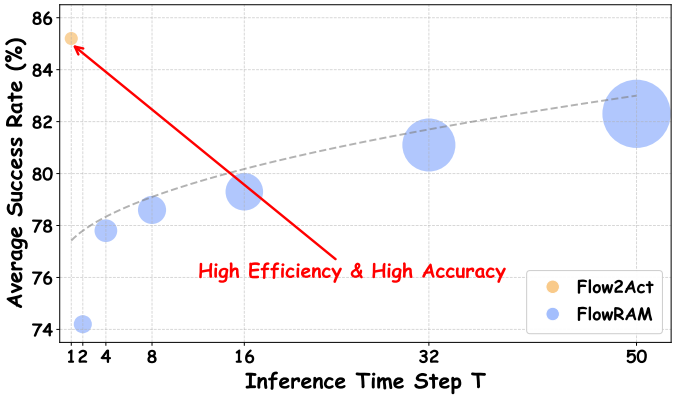

Fig. 6. **Comparison of success rate and efficiency.** We compare Flow2Act with FlowRAM across a few-shot setting. Flow2Act achieves superior accuracy in a single inference step, consistently outperforming FlowRAM at all timesteps. Bubble area indicates the inference time of the model.

One-step Inference vs. Multi-step Generation. A critical advantage of Flow2Act is its ability to achieve superior performance with single-step inference, which fundamentally differs from conventional diffusion-based policies that require multiple denoising steps [34], [47]. As demonstrated in Fig. 6, Flow2Act achieves 85.0% average success rate with only one inference step, consistently outperforming FlowRAM across all time steps. In contrast, FlowRAM requires approximately 32 inference steps to reach a comparable success rate, demonstrating the substantial computational overhead of multi-step generation methods. This single-step capability is made possible by our MeanFlow objective, which reparameterizes the policy dynamics as the mean velocity field defined over a time interval. Unlike traditional flow matching methods that require numerical ODE solvers, our approach enables direct mapping from noise to the target action through a closed-form update, eliminating the need for iterative integration. The result is a method that simultaneously achieves high computational efficiency and superior manipulation accuracy, demonstrating that the trade-off between inference speed and performance can be overcome when the training objective is properly aligned with the requirements of robotic control.

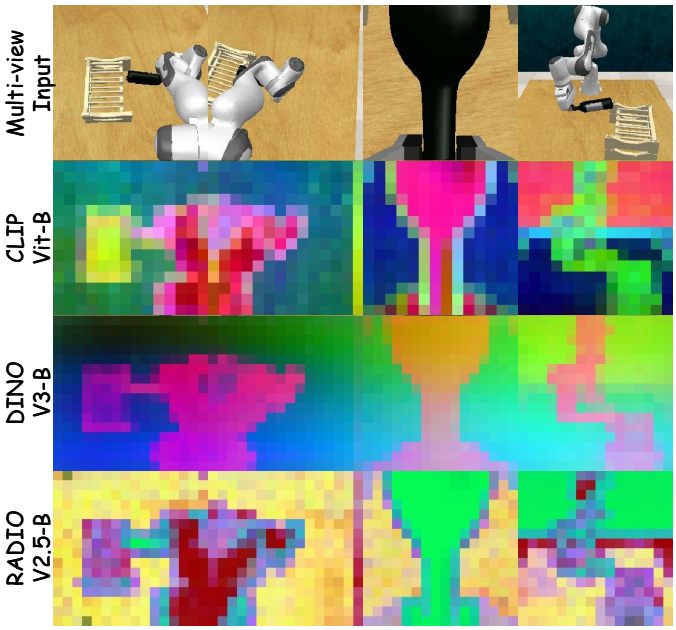

Fig. 7. **PCA feature visualizations of CLIP, DINOV3-B, and RADIOV2.5-B models under multi-view settings.**

PCA Visualization of Feature Representations. The PCA visualization in Fig. 7 reveals critical differences in feature representations across vision models. CLIP [54] features exhibit significant multi-view inconsistency, with the same object (*e.g.*, rack) represented differently across camera perspectives. This semantic inconsistency limits cross-view correspondence for precise manipulation. DINO [58] fails to effectively separate background regions from task-critical foreground elements, with feature responses spreading indiscriminately across both object and background areas. This lack of spatial selectivity compromises contact-scale accuracy. In contrast, RADIOv2.5 [27] demonstrates superior representation capabilities, featuring cross-view consistency and precise foreground-background separation. Its feature maps exhibit sharp boundaries that enable pixel-level spatial reasoning. The core advantage resides in its ability to balance semantic consistency with fine-grained spatial perception, overcoming

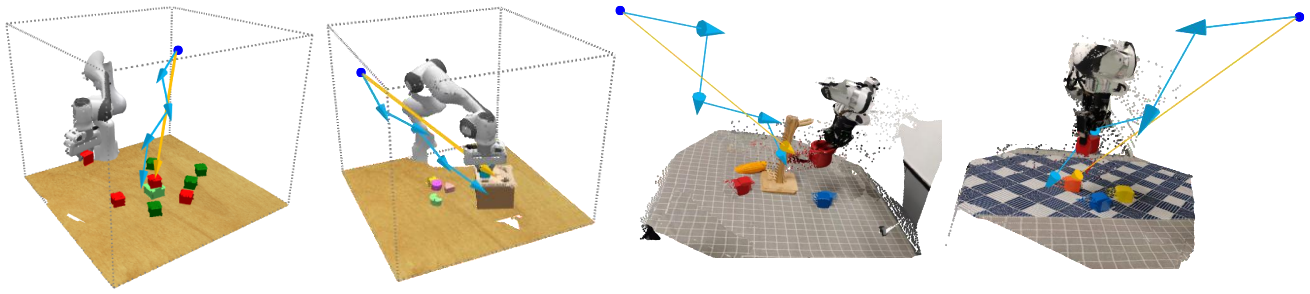

Fig. 8. **Trajectory generation via flow matching under two velocity formulations.** **Blue arrows** represent trajectories generated using instantaneous velocity flow matching; **yellow arrows** correspond to those generated using average velocity flow matching. Each panel illustrates the robot's end-effector motion in distinct manipulation scenarios, with trajectories visualized as directed segments from initial to goal poses.

fundamental limitations of prior approaches and ensuring robust manipulation performance under visual variations.

D. Real-Robot Experiments

Environmental Setup. As illustrated in Fig. 5, we evaluate our method on six distinct physical manipulation tasks and compare it against the previous state-of-the-art approach. All experiments are conducted using a Franka Research 3 robotic arm equipped with a Robotiq 2F-85 gripper, mounted on a fixed tabletop setup. For visual perception, we employ two calibrated Intel RealSense D435i RGB-D cameras, positioned to the left and right of the robot, to provide stereoscopic coverage of the workspace. Each task involves 20 expert demonstrations. Object configurations are randomized across demonstrations to enhance the diversity of training data and prevent overfitting to specific arrangements. Task variants include multiple object types and instruction formulations to evaluate the system's adaptability. For inference, we apply the BiRRT planner integrated with MoveIt! ROS package [85] to reach the predicted action poses.

We design two distinct evaluation settings to comprehensively assess the system's capabilities: **Basic Setting**: This evaluation protocol assesses the fundamental performance of the system without any environmental perturbations. In this setting, we maintain consistent background, lighting conditions, object appearance, and camera viewpoints to establish a baseline for the method's core functionality. **Generalization Setting**: To evaluate the system's robustness and adaptability, we introduce six types of environmental perturbations during evaluation: (1) `Background variations`: Changing the scene background while keeping the task objects consistent; (2) `Distractor objects`: Adding irrelevant objects to the workspace to test the system's ability to focus on task-relevant elements; (3) `Spatial location variations`: Randomizing the starting positions of objects within the workspace; (4) `Object appearance changes`: Using objects with different colors, textures, or slight shape variations; (5) `Camera view adjustments`: Modifying the camera angles and positions within reasonable limits; (6) `Lighting conditions`: Altering the intensity and direction of lighting to simulate different environmental conditions. Training proceeds for 30k steps on the collected real-world dataset with visual augmentation, followed by an additional 10k steps of fine-tuning. All hyperparameters are kept consistent with the simulation experiments to demonstrate direct transferability of our approach.

TABLE VI
REAL-WORLD PERFORMANCE IN BASIC AND GENERALIZATION SETTING.

	Basic Setting		Generalization Setting	
Task / Method	FlowRAM	Flow2Act	FlowRAM	Flow2Act
Stack Blocks	8/10	**10/10**	32/60	**51/60**
Hanging Cup	5/10	**7/10**	24/60	**37/60**
Put Mustard	7/10	**8/10**	33/60	**42/60**
Stack cups	6/10	**8/10**	30/60	**42/60**
Put in Drawer	10/10	**10/10**	46/60	**53/60**
Sort Shape	7/10	**8/10**	34/60	**40/60**
Average	43/60	**51/60**	199/360	**265/360**

Quantitative Results. In Table VI, we report the average success rates across six real-world manipulation tasks under both basic and generalization settings. The results demonstrate that Flow2Act consistently outperforms FlowRAM across all evaluated tasks and settings. Notably, Flow2Act achieves an average success rate of 51/60 in the basic setting, significantly surpassing FlowRAM's 43/60. The most substantial improvements are observed in the generalization setting, where Flow2Act attains an impressive success rate of 265/360 compared to FlowRAM's 199/360. On tasks involving distractor objects, FlowRAM struggles to distinguish between different unseen items, leading to frequent failure cases. In contrast, Flow2Act maintains robust performance, benefiting from the pretrained spatially grounded and semantically coherent representations that enable effective object differentiation under cluttered conditions. Additionally, in long-horizon tasks such as Stack Cups, Flow2Act achieves sizable performance gains, which we attribute to the joint learning of 3D geometry and future dynamics during the pretraining stage, facilitating effective reasoning and manipulation in physically complex scenarios. The consistent outperformance across all tasks confirms the effectiveness of Flow2Act's integrated approach to agglomerative perception and one-step action generation for real-world robotic manipulation.

E. Action Generation process

As illustrated in Fig. 8, when limited to only four time steps, flow matching based on instantaneous velocity fails to produce accurate keyframe poses. In contrast, our approach, which employs flow matching over average velocity fields, yields significantly more precise and geometrically consistent pose predictions. This improvement stems from the fact that average velocity encodes smoother, more temporally coherent motion priors, thereby reducing oscillatory artifacts and promoting stable convergence. Our results demonstrate that modeling motion dynamics through averaged velocity fields better aligns with the underlying physical constraints of robotic manipulation, leading to more reliable trajectory generation under sparse sampling regimes.

V. Conclusion and Discussion

In this work, we introduced Flow2Act, a unified visuomotor policy framework designed to resolve the tripartite conflict between generalization, inference latency, and manipulation precision. By synergizing an agglomerative multi-teacher vision backbone with a reparameterized mean-velocity flow generator, Flow2Act establishes a new paradigm for efficient, geometry-aware robotic control. Our extensive empirical evaluations on RLBench and Colosseum benchmarks demonstrate that removing task-specific visual pretraining in favor of distilled foundation models significantly enhances robustness against severe visual disturbances. Furthermore, the proposed Conditional MeanFlow objective successfully reduces the inference cost of generative policies to a single function evaluation without compromising the expressivity required for multimodal action distributions. The integration of a curriculum-based region-aware mechanism further ensures that this efficiency does not come at the cost of contact-level accuracy, effectively bridging the gap between semantic understanding and fine-grained physical interaction.

References

[1] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response," IEEE TPAMI, vol. 38, no. 1, pp. 14–29, 2015.

[2] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," in RSS, 2023.

[3] Y. Chen, Y. Geng, F. Zhong, J. Ji, J. Jiang, Z. Lu, H. Dong, and Y. Yang, "Bi-dexhands: Towards human-level bimanual dexterous manipulation," IEEE TPAMI, vol. 46, no. 5, pp. 2804–2818, 2023.

[4] Z. Yang, N. Song, W. Li, X. Zhu, L. Zhang, and P. H. Torr, "Deepinteraction++: Multi-modality interaction for autonomous driving," IEEE TPAMI, 2025.

[5] G. R. Team, S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, T. Armstrong, A. Balakrishna, R. Baruch, M. Bauza, M. Blokzijl et al., "Gemini robotics: Bringing ai into the physical world," arXiv preprint arXiv:2503.20020, 2025.

[6] X. Wang, Q. Huang, A. Celikyilmaz, J. Gao, D. Shen, Y.-F. Wang, W. Y. Wang, and L. Zhang, "Vision-language navigation policy learning and adaptation," IEEE TPAMI, vol. 43, no. 12, pp. 4205–4216, 2020.

[7] Z. Wang, S. Cai, A. Liu, Y. Jin, J. Hou, B. Zhang, H. Lin, Z. He, Z. Zheng, Y. Yang et al., "Jarvis-1: Open-world multi-task agents with memory-augmented multimodal language models," IEEE TPAMI, 2024.

[8] Q. Bu, Y. Yang, J. Cai, S. Gao, G. Ren, M. Yao, P. Luo, and H. Li, "Univla: Learning to act anywhere with task-centric latent actions," RSS, 2025.

[9] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter et al., "pi_0: A vision-language-action flow model for general robot control," arXiv preprint arXiv:2410.24164, 2024.

[10] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang et al., "Gr00t n1: An open foundation model for generalist humanoid robots," arXiv preprint arXiv:2503.14734, 2025.

[11] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong et al., "Openvla: An open-source vision-language-action model," in CoRL. PMLR, 2025, pp. 2679–2713.

[12] Q. Zhao, Y. Lu, M. J. Kim, Z. Fu, Z. Zhang, Y. Wu, Z. Li, Q. Ma, S. Han, C. Finn et al., "Cot-vla: Visual chain-of-thought reasoning for vision-language-action models," in CVPR, 2025, pp. 1702–1713.

[13] J. Wen, Y. Zhu, M. Zhu, Z. Tang, J. Li, Z. Zhou, X. Liu, C. Shen, Y. Peng, and F. Feng, "Diffusionvla: Scaling robot foundation models via unified diffusion and autoregression," in ICML, 2025.

[14] J. Liu, M. Liu, Z. Wang, L. Lee, K. Zhou, P. An, S. Yang, R. Zhang, Y. Guo, and S. Zhang, "Robomamba: Efficient vision-language-action model for robotic reasoning and manipulation," NeurIPS, 2024.

[15] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain et al., "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0," in ICRA. IEEE, 2024, pp. 6892–6903.

[16] H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du et al., "Bridgedata v2: A dataset for robot learning at scale," in CoRL. PMLR, 2023, pp. 1723–1736.

[17] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in ICRA, vol. 2. IEEE, 2000, pp. 995–1001.

[18] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in CoRL. PMLR, 2023, pp. 785–799.

[19] Y. Ze, G. Yan, Y.-H. Wu, A. Macaluso, Y. Ge, J. Ye, N. Hansen, L. E. Li, and X. Wang, "Gnfactor: Multi-task real robot learning with generalizable neural feature fields," in CoRL. PMLR, 2023, pp. 284–301.

[20] A. Goyal, V. Blukis, J. Xu, Y. Guo, Y.-W. Chao, and D. Fox, "Rvt2: Learning precise manipulation from few demonstrations," RSS, 2024.

[21] Y. Mu, T. Chen, Z. Chen, S. Peng, Z. Lan, Z. Gao, Z. Liang, Q. Yu, Y. Zou, M. Xu et al., "Robotwin: Dual-arm robot benchmark with generative digital twins," in CVPR, 2025, pp. 27 649–27 660.

[22] F. Lin, Y. Hu, P. Sheng, C. Wen, J. You, and Y. Gao, "Data scaling laws in imitation learning for robotic manipulation," ICLR, 2024.

[23] 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.

[24] J. Gao, A. Xie, T. Xiao, C. Finn, and D. Sadigh, "Efficient data collection for robotic manipulation via compositional generalization," arXiv preprint arXiv:2403.05110, 2024.

[25] S. Qian, K. Mo, V. Blukis, D. F. Fouhey, D. Fox, and A. Goyal, "3d-mvp: 3d multiview pretraining for manipulation," in CVPR, 2025, pp. 22 530–22 539.

[26] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell, "Real-world robot learning with masked visual pre-training," in CoRL. PMLR, 2023, pp. 416–426.

[27] G. Heinrich, M. Ranzinger, H. Yin, Y. Lu, J. Kautz, A. Tao, B. Catanzaro, and P. Molchanov, "Radiov2. 5: Improved baselines for agglomerative vision foundation models," in CVPR, 2025, pp. 22 487–22 497.

[28] Y. Avigal, L. Berscheid, T. Asfour, T. Kröger, and K. Goldberg, "Speedfolding: Learning efficient bimanual folding of garments," in IROS. IEEE, 2022, pp. 1–8.

[29] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," CoRL, 2021.

[30] J. Wu, X. Sun, A. Zeng, S. Song, J. Lee, S. Rusinkiewicz, and T. Funkhouser, "Spatial action maps for mobile manipulation," RSS, 2020.

[31] D. Qu, H. Song, Q. Chen, Y. Yao, X. Ye, Y. Ding, Z. Wang, J. Gu, B. Zhao, D. Wang et al., "Spatialvla: Exploring spatial representations for visual-language-action model," RSS, 2025.

[32] X. Liu, C. Gong et al., "Flow straight and fast: Learning to generate and transfer data with rectified flow," in ICLR, 2023.

[33] R. T. Chen and Y. Lipman, "Flow matching on general geometries," in ICLR, 2024.

[34] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," IJRR, p. 02783649241273668, 2023.

[35] X. Ma, S. Patidar, I. Haughton, and S. James, "Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation," in CVPR, 2024, pp. 18 081–18 090.

[36] T. Oba, M. Walter, and N. Ukita, "Read: Retrieval-enhanced asymmetric diffusion for motion planning," in CVPR, 2024, pp. 17 974–17 984.

[37] Z. Xian, N. Gkanatsios, T. Gervet, T.-W. Ke, and K. Fragkiadaki, "Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation," in CoRL, 2023.

[38] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, "Diffusion models in vision: A survey," IEEE TPAMI, vol. 45, no. 9, pp. 10 850–10 869, 2023.

[39] J. Ma, X. Chen, W. Bao, J. Xu, and H. Wang, "Madiff: Motion-aware mamba diffusion models for hand trajectory prediction on egocentric videos," IEEE TPAMI, 2024.

[40] H. Zhang, Z. Wang, D. Zeng, Z. Wu, and Y.-G. Jiang, "Diffusionad: Norm-guided one-step denoising diffusion for anomaly detection," IEEE TPAMI, 2025.

[41] S. Wang, L. Wang, S. Zhou, J. Tian, J. Li, H. Sun, and W. Tang, "Flowram: Grounding flow matching policy with region-aware mamba framework for robotic manipulation," in CVPR, 2025, pp. 12 176–12 186.

[42] Q. Zhang, Z. Liu, H. Fan, G. Liu, B. Zeng, and S. Liu, "Flowpolicy: Enabling fast and robust 3d flow-based policy via consistency flow matching for robot manipulation," in AAAI, vol. 39, no. 14, 2025, pp. 14 754–14 762.

[43] X. Hu, B. Liu, X. Liu, and Q. Liu, "Adaflow: Imitation learning with variance-adaptive flow-based policies," NeurIPS, 2024.

[44] F. Zhang and M. Gienger, "Affordance-based robot manipulation with flow matching," arXiv preprint arXiv:2409.01083, 2024.

[45] E. Chisari, N. Heppert, M. Argus, T. Welschehold, T. Brox, and A. Valada, "Learning robotic manipulation policies from point clouds with conditional flow matching," in CoRL, 2024.

[46] Q. Rouxel, A. Ferrari, S. Ivaldi, and J.-B. Mouret, "Flow matching imitation learning for multi-support manipulation," in 2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids). IEEE, 2024, pp. 528–535.

[47] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki, "3d diffuser actor: Policy diffusion with 3d scene representations," CoRL, 2024.

[48] S. James, Z. Ma, D. Rovick Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," IEEE Robotics and Automation Letters, 2020.

[49] W. Pumacay, I. Singh, J. Duan, R. Krishna, J. Thomason, and D. Fox, "The colosseum: A benchmark for evaluating generalization for robotic manipulation," RSS, 2024.

[50] T. Dao and A. Gu, "Transformers are ssms: Generalized models and efficient algorithms through structured state space duality," ICML, 2024.

[51] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," Journal of Machine Learning Research, vol. 17, no. 39, pp. 1–40, 2016.

[52] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "Bc-z: Zero-shot task generalization with robotic imitation learning," in CoRL. PMLR, 2022, pp. 991–1002.

[53] S. James, K. Wada, T. Laidlow, and A. J. Davison, "Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation," in CVPR, 2022, pp. 13 739–13 748.

[54] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark et al., "Learning transferable visual models from natural language supervision," in ICML. PMLR, 2021, pp. 8748–8763.

[55] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," in CoRL. PMLR, 2022, pp. 894–906.

[56] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in ICCV, 2023, pp. 11 975–11 986.

[57] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby et al., "Dinov2: Learning robust visual features without supervision," arXiv preprint arXiv:2304.07193, 2023.

[58] O. Siméoni, H. V. Vo, M. Seitzer, F. Baldassarre, M. Oquab, C. Jose, V. Khalidov, M. Szafraniec, S. Yi, M. Ramamonjisoa et al., "Dinov3," arXiv preprint arXiv:2508.10104, 2025.

[59] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo et al., "Segment anything," in ICCV, 2023, pp. 4015–4026.

[60] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi et al., "OpenVLA: An open-source vision-language-action model," CoRL, 2024.

[61] M. J. Kim, C. Finn, and P. Liang, "Fine-tuning vision-language-action models: Optimizing speed and success," NeurIPS, 2024.

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

[63] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," NeurIPS, vol. 33, pp. 6840–6851, 2020.

[64] S. Li, Y. Qin, M. Zheng, X. Jin, and Y. Liu, "Diff-bgm: A diffusion model for video background music generation," in CVPR, 2024, pp. 27 348–27 357.

[65] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel et al., "Scaling rectified flow transformers for high-resolution image synthesis," in ICML, 2024.

[66] S. Yan, Z. Zhang, M. Han, Z. Wang, Q. Xie, Z. Li, Z. Li, H. Liu, X. Wang, and S.-C. Zhu, "M2diffuser: Diffusion-based trajectory optimization for mobile manipulation in 3d scenes," IEEE TPAMI, 2025.

[67] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," in RSS, 2024.

[68] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in ICLR, 2023.

[69] Q. Dao, H. Phung, B. Nguyen, and A. Tran, "Flow matching in latent space," arXiv preprint arXiv:2307.08698, 2023.

[70] A. Davtyan, S. Sameni, and P. Favaro, "Efficient video prediction via sparsely conditioned flow matching," in ICCV, 2023, pp. 23 263–23 274.

[71] Q. Zheng, M. Le, N. Shaul, Y. Lipman, A. Grover, and R. T. Chen, "Guided flows for generative modeling and decision making," arXiv preprint arXiv:2311.13443, 2023.

[72] Z. Geng, M. Deng, X. Bai, J. Z. Kolter, and K. He, "Mean flows for one-step generative modeling," NeurIPS, 2025.

[73] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, and D. Fox, "Rvt: Robotic view transformer for 3d object manipulation," in CoRL. PMLR, 2023, pp. 694–710.

[74] G. Lu, S. Zhang, Z. Wang, C. Liu, J. Lu, and Y. Tang, "Manigaussian: Dynamic gaussian splatting for multi-task robotic manipulation," ECCV, 2024.

[75] P.-L. Guhur, S. Chen, R. G. Pinel, M. Tapaswi, I. Laptev, and C. Schmid, "Instruction-driven history-aware policies for robotic manipulations," in CoRL. PMLR, 2023, pp. 175–187.

[76] T. Gervet, Z. Xian, N. Gkanatsios, and K. Fragkiadaki, "Act3d: 3d feature field transformers for multi-task robotic manipulation," in CoRL, 2023.

[77] D. Liang, X. Zhou, W. Xu, X. Zhu, Z. Zou, X. Ye, X. Tan, and X. Bai, "Pointmamba: A simple state space model for point cloud analysis," in NeurIPS, 2024.

[78] Q. Li, Y. Liang, Z. Wang, L. Luo, X. Chen, M. Liao, F. Wei, Y. Deng, S. Xu, Y. Zhang et al., "Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation," arXiv preprint arXiv:2411.19650, 2024.
[79] S. Wang, J. You, Y. Hu, J. Li, and Y. Gao, "Skil: Semantic keypoint imitation learning for generalizable data-efficient manipulation," RSS, 2025.
[80] M. Ranzinger, G. Heinrich, J. Kautz, and P. Molchanov, "Am-radio: Agglomerative vision foundation model reduce all domains into one," in CVPR, 2024, pp. 12 490–12 500.
[81] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," in First conference on language modeling, 2024.
[82] I. Loshchilov, "Decoupled weight decay regularization," arXiv preprint arXiv:1711.05101, 2017.
[83] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," CoRL, 2022.
[84] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik, "Masked visual pre-training for motor control," arXiv preprint arXiv:2203.06173, 2022.
[85] D. Coleman, I. Sucan, S. Chitta, and N. Correll, "Reducing the barrier to entry of complex robotic software: a moveit! case study," arXiv preprint arXiv:1404.3785, 2014.

Sen Wang received the B.E. degree in Control Science and Engineering from Jilin University, Changchun, China, in 2023. He is currently a third-year Ph.D. candidate in the Institute of Artificial Intelligence and Robotics at Xi'an Jiaotong University. His research interests include embodied computer vision, robotic manipulation and interactive world model. He has published four papers in CVPR and NIPS.

Le Wang (Senior Member, IEEE) received the B.S. and Ph.D. degrees in control science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2008 and 2014, respectively. From 2013 to 2014, he was a visiting Ph.D. student with the Stevens Institute of Technology, Hoboken, NJ, USA. From 2016 to 2017, he was a visiting scholar with Northwestern University, Evanston, IL, USA. He is currently a professor with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. His research interests include computer vision, pattern recognition, and machine learning. He is an associate editor for PR, MVA, and PRL.

Sanping Zhou (Member, IEEE) received the Ph.D. degree in control science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2020. From 2018 to 2019, he was a visiting Ph.D. student with Robotics Institute, Carnegie Mellon University. He is currently a professor with the Institute of Artificial Intelligence and Robotics, Xian Jiaotong University. His research interests include machine learning, deep learning, and computer vision, with a focus on personre-identification, salient object detection, medical image segmentation, image classification, and visual tracking.

Kun Xia received the Ph.D. degree in Control Science and Engineering from Xi'an Jiaotong University, Xi'an, China, in 2024. From 2022 to 2023, he was a visiting Ph.D. student with University of Illinois Chicago, IL, USA. He is currently an Assistant Professor with the School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, China. His research interests include computer vision, image/video processing, analysis and understanding.

Gang Hua (Fellow, IEEE) received the B.S. and M.S. degrees in automatic control engineering from Xi'an Jiaotong University, Xi'an, China, in 1999 and 2002, respectively, and the Ph.D. degree in electrical engineering and computer science from Northwestern University, Evanston, IL, USA, in 2006. He was a senior scientist with Microsoft Live labs Research from 2006 to 2009, a senior researcher with Nokia Research Center Hollywood from 2009 to 2010, a research staff member with IBM Research T. J. Watson Center from 2010 to 2011, and a visiting researcher from 2011 to 2014. During 2014–15, he took a leave and worked on the Amazon-Go project. He was an associate professor with the Stevens Institute of Technology from 2011 to 2015. He was also with Microsoft from 2015 to 2018 as the Science/Technical adviser to the CVP of the Computer Vision Group, director of Computer Vision Science Team in Redmond, Taipei ATL, and senior principal researcher/research manager with Microsoft Research. He was the CTO with Convenience Bee, and the managing director and chief scientist of its research branch in US, Wormpex AI Research, from 2018 to 2024. He was the vice president with Multimodal Experiences Research Lab, Dolby Laboratories from 2024 to 2025. He is currently the director of applied science with Amazon Alexa AI. His research interests include computer vision, pattern recognition, machine learning, robotics, towards general artificial intelligence, with primary applications in cloud and edge intelligence. He is an associate editor for TPAMI and MVA. He is a general chair of ICCV'2027 and a program chair of CVPR'2019 & 2022. He is the recipient of the 2015 IAPR Young Biometrics Investigator Award. He is an IAPR Fellow and an ACM distinguished scientist.