# MOTION INTO ACTION: FROM OBJECT-CENTRIC PERCEPTION TO DYNAMICS-CENTRIC ACTION GENERATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Vision-language-action (VLA) models perform well in static manipulation but remain challenged by independently moving targets. As objects continue to move during inference, generated actions can become misaligned with the scene at execution time. Existing approaches address this perception-to-execution gap through execution scheduling or scene modeling, but often leave observed target dynamics weakly coupled to action generation. We propose *Motion Into Action* (MIA), a framework that couples object-centric perception with dynamics-conditioned action generation. MIA maps task-relevant optical-flow history to a motion-induced latent action that parameterizes a motion-aligned Gaussian source for diffusion, guiding the action head toward executable actions in fewer denoising steps. Context-aware asynchronous execution further overlaps inference and execution while promoting temporal consistency across action chunks. Across five DOMINO tasks, MIA achieves an average success rate of 41.6%, relatively outperforming the strongest baseline by 55.22%. On two tasks, it maintains performance when denoising is reduced from 20 to 5 steps. Across five real-world tasks, MIA achieves an average success rate of 88%, with qualitative examples illustrating transfer to faster motion, occlusion, and moving distractors beyond the demonstration conditions.

## 1 INTRODUCTION

Grasping a moving object requires coupling the agent's own movement to the target's ongoing motion. This coordination emerges early in human development: studies of infant catching suggest that reaches are organized relative to the moving object, with the hand moving with the target while approaching it (von Hofsten, 1983). Such behavior illustrates perception–action coupling, in which task-relevant visual information guides the organization of movement (Warren, 2006). Despite recent progress in vision-language-action (VLA) models (Black et al., 2025; NVIDIA et al., 2025; Kim et al., 2025), maintaining this coupling remains challenging for independently moving targets. While a policy interprets an observation and generates an action, the target continues moving, creating a perception-to-execution gap that can leave the action misaligned with the target at execution. Our premise is that observed object motion should shape the construction of actions themselves, rather than enter only as auxiliary scene context. For diffusion-based policies, this means converting task-relevant motion into an action prior that determines where generation begins, enabling refinement of a motion-aligned response without explicit future-scene prediction.

To mitigate this perception-to-execution gap, prior work has pursued three complementary directions. Asynchronous VLA systems (Xie et al., 2026; Tang et al., 2025; Black et al., 2026) generate the next action chunk while executing the current one. When execution time exceeds inference time, this overlap hides much of the inference latency, reduces inter-chunk idle time, and allows the policy to react to more recent observations. Action-streaming methods (Høeg et al., 2025; Lu et al., 2026; Chen et al., 2025) treat an action sequence as a continuously updated stream. They maintain a rolling, partially refined buffer: leading actions are sufficiently denoised for execution, while later actions remain at higher noise levels or earlier sampling stages. As execution advances, the clear action dequeue while the new pure noise enqueue. By overlapping execution with sampling and reusing intermediate computation, these methods reduce the time required to obtain the next



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

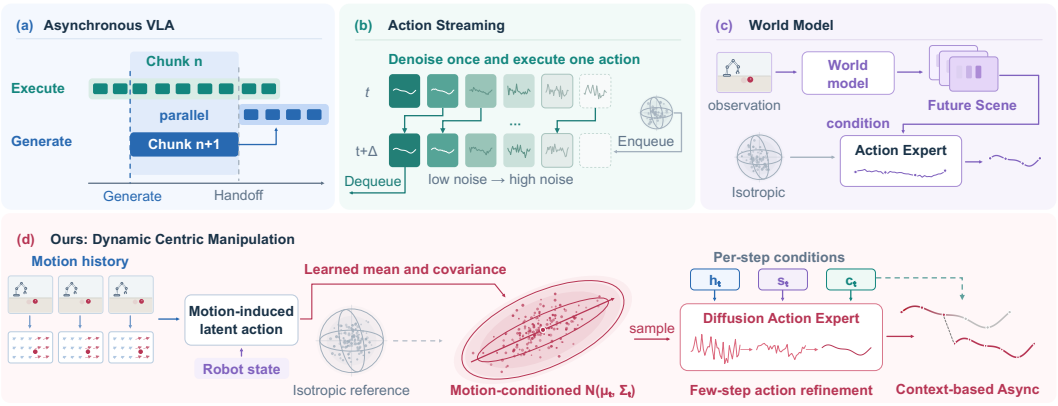



Figure 1: **Comparison of approaches to dynamic manipulation.**

executable action. Dynamics-aware world/action models (Lou et al., 2026; Syed et al., 2026; Chen et al., 2026) address observation staleness by modeling how the target and scene evolve over time. They encode observed motion or predict future latent states and provide these representations to the action expert, allowing generated actions to better match the scene expected at execution time.

However, these approaches primarily improve action availability or enrich policy conditioning. Asynchronous inference and action streaming reduce waiting through scheduling and reuse, without necessarily shortening refinement of the full action horizon. World models provide predicted scene context, while object-centric methods (Wang et al., 2026a; Kambara et al., 2026; Hanyu et al., 2026) strengthen target grounding and may also encode temporal dynamics. For diffusion-based policies, richer conditioning alone does not make the action source motion-dependent. The key opportunity is to translate observed target motion directly into an action prior, so that generation begins from a motion-aligned response and refines it into executable actions without future-scene prediction.

To close this gap, we propose *Motion Into Action* (MIA), a motion-prior-guided asynchronous action-generation framework that couples observed object dynamics directly to efficient diffusion-based action expert. MIA extracts task-relevant target motion from short-term optical-flow history and translates it into a motion-induced latent action, which represents the coarse robot response implied by the target's motion rather than a complete executable trajectory. Instead of starting every generation process from a scene-agnostic isotropic Gaussian, MIA uses this latent action to parameterize a sample-specific, motion-aligned source distribution for the diffusion action expert. Conditioned on the current vision-language context and robot state, the action expert therefore begins from an action region already aligned with the target dynamics and focuses its denoising capacity on completing precise pose, contact, and embodiment constraints. For continuous execution, MIA predicts the next action chunk while the robot executes the current one and conditions the new chunk on recently committed actions, reducing idle time while preserving temporal consistency across chunk boundaries. By aligning the starting point of denoising with target dynamics while overlapping action generation with execution, MIA is designed to produce executable actions with fewer denoising steps without explicitly forecasting future images or reconstructing object trajectories.

We evaluate MIA on the DOMINO benchmark (Fang et al., 2026) across 5 simulation tasks and on 5 real-world tasks. In simulation, MIA achieves an average success rate of $41.6\%$ and a manipulation score of 51.97, attaining state-of-the-art performance. When reducing denoising steps from 20 to 5 steps in *adjust_bottle* and *handover_mic* tasks, MIA achieves 119.50% success rate, whereas the previous state-of-the-art method, VLASH (Tang et al., 2025), retains only 60.61%. In real-world tasks, MIA achieves 88% success rate, adapts its actions to changes in target speed and direction, demonstrating real-time responsiveness under non-stationary object motion.

## 2 Related Work

### 2.1 Dynamic Robotic Manipulation

Dynamic manipulation requires policies to account for how a target evolves during perception, action generation, and execution. Existing approaches broadly follow three directions. Motion-aware poli-

cies learn temporal representations from observation histories or dynamic demonstrations (Zhang et al., 2025; Fang et al., 2026). Responsiveness-oriented systems reduce delay through efficient inference, asynchronous action generation and execution, or continuously refined action streams (Xie et al., 2026; Ma et al., 2025; Black et al., 2026; Tang et al., 2025; Høeg et al., 2025; Chen et al., 2025; Lu et al., 2026). Prediction-based methods instead compensate for stale observations by correcting generated action chunks or forecasting future object and scene representations (Zhang et al., 2026; Wang et al., 2026a; Syed et al., 2026). These approaches improve either responsiveness or anticipation, but generally leave target-motion understanding and efficient action generation as separate mechanisms. In contrast, MIA uses observed target motion to shape where diffusion-based action generation begins while retaining asynchronous execution, without an explicit future-state rollout or complete object-trajectory reconstruction.

### 2.2 Optical and Scene Flow for Robotic Manipulation

Optical flow describes dense two-dimensional inter-frame motion, while scene flow extends this representation to three dimensions. Both expose motion direction, magnitude, and spatial support without requiring explicit object trajectories, but neither inherently separates task-relevant target motion from camera, robot, or background motion. In robotic manipulation, learned motion fields have been used to generate trajectories, predict future object or scene states, and condition action policies on scene dynamics (Kambara et al., 2026; Fang et al., 2026; Syed et al., 2026; Wang et al., 2026a;b). These methods establish optical and scene flow as useful intermediate representations, but typically treat them as prediction targets, trajectory outputs, or auxiliary conditions. MIA instead extracts task-relevant motion from short-term optical-flow history, maps it to a motion-induced latent action, and uses that latent action to parameterize the diffusion source distribution. The motion representation therefore determines the starting region of action generation rather than directly specifying a final trajectory or future scene.

### 2.3 Source Priors for Generative Action Policies

Generative action policies commonly transform samples from an observation-independent Gaussian source into executable actions through diffusion or flow dynamics. Most real-time methods retain this source and improve action availability by redistributing, scheduling, or reusing the subsequent denoising process (Høeg et al., 2025; Black et al., 2026; Chen et al., 2025; Lu et al., 2026). A complementary line of work learns structured or state-dependent source priors from recent actions, robot proprioception, or latent motion modes (Kang et al., 2026; Dai et al., 2026; Machado et al., 2026). Together, these studies identify the source distribution as a design axis distinct from the denoiser architecture and sampling solver. MIA differs in how this source is constructed: its prior is continuous, sample-specific, and induced by the currently observed target motion rather than only by robot state, action history, or dataset-level motion patterns. Recently committed actions are used separately to preserve consistency across asynchronous chunks.

## 3 Approach

To address the coupled challenges of observations and action-generation latency, MIA integrates motion-aligned source distribution with asynchronous execution. As illustrated in figure 2, MIA first extracts task-relevant target motion from short-term optical-flow history and maps it to a motion-induced latent action that represents the coarse robot response implied by the observed dynamics. This latent action parameterizes a sample-specific source distribution, from which the diffusion action expert refines an executable action chunk. In parallel, the next chunk is generated while the current chunk is being executed, with recently committed actions providing temporal context across chunk boundaries to achieve fast reaction and smooth action. We first formalize the task and overall pipeline (Section 3.1), then describe motion-induced latent-action extraction (Section 3.2), motion-aligned source-based action generation (Section 3.3) and asynchronous execution (Section 3.4).

### 3.1 Problem Formulation and Method Overview

At control time $t$, the robot receives a language instruction $\ell$, its current proprioceptive state $\mathbf{s}_t$, and a short history of RGB observations $\mathcal{O}_t$ from $V$ camera views. Optical flow $\mathcal{F}_t$ is computed only

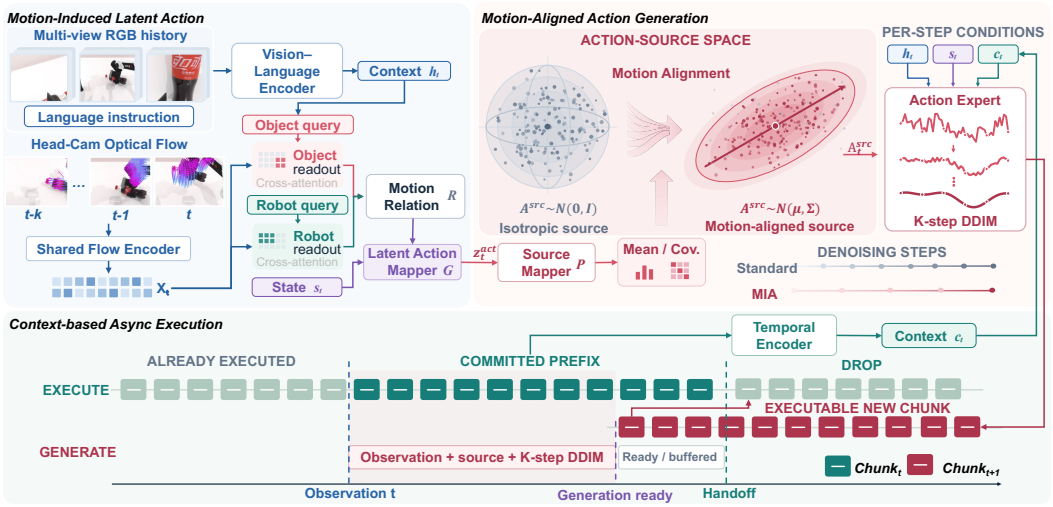



Figure 2: **Method Overview of MIA.** The framework comprises three components: (i) motion-induced latent-action extraction from task-relevant optical-flow features and proprioception; (ii) motion-aligned Gaussian source parameterization for few-step action generation; and (iii) asynchronous execution with action-context conditioning for temporal consistency.

from the fixed head camera. Since the head camera remains fixed during manipulation, its flow captures the image-plane motion of the robot arms and objects in a consistent coordinate frame. Given $(\mathcal{O}_t, \mathcal{F}_t, \ell, \mathbf{s}_t)$, the policy predicts an action chunk

$$\mathbf{A}_t = [\mathbf{a}_{t,0}, \ldots, \mathbf{a}_{t,H-1}] \in \mathbb{R}^{H \times d_a} \tag{1}$$

where $H$ is the action horizon and $d_a$ is the action dimension. The subscript $t$ identifies the observation used for prediction, while the second subscript indexes actions within the chunk. Its execution begins at the handoff determined by the asynchronous schedule in Section 3.4. The target may therefore move between the latest observation and the execution of $\mathbf{A}_t$. Our goal is to use recent object's target motion to generate an action chunk efficiently, without explicitly forecasting future images or reconstructing a three-dimensional object trajectory.

We use $t$ for environment time and $k \in \{0, \ldots, K\}$ for the diffusion step. A conventional diffusion action expert initializes every sample from the same motion-agnostic source, $\mathbf{A}_{t,K} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and iteratively maps it to $\mathbf{A}_{t,0}$. In contrast, MIA first forms a motion-induced latent action and then uses it to determine a sample-specific source:

$$\mathbf{z}_t^{\text{motion}} = E(\mathcal{F}_t, \mathbf{h}_t)\,, \qquad \mathbf{z}_t^{\text{act}} = G\big(\mathbf{z}_t^{\text{motion}}, \mathbf{s}_t\big) \tag{2}$$

where $\mathbf{h}_t$ denotes the vision-language features of the multi-view RGB observations and instruction. The encoder $E$ uses this context to extract the task-relevant motion representation $\mathbf{z}_t^{\text{motion}}$ from $\mathcal{F}_t$. The mapper $G$ combines it with the robot state $\mathbf{s}_t$ to produce $\mathbf{z}_t^{\text{act}}$, which encodes the coarse robot response implied by the observed motion.

This latent action then parameterizes the diffusion source, providing a motion-aligned initialization for generating an executable action chunk:

$$\mathbf{A}_t^{\text{src}} \sim p\big(\mathbf{A}^{\text{src}} \mid \mathbf{z}_t^{\text{act}}, \mathbf{s}_t\big)\,, \qquad \widehat{\mathbf{A}}_t = D^{(K)}(\mathbf{A}_t^{\text{src}} \mid \mathbf{h}_t, \mathbf{s}_t, \mathbf{c}_t) \tag{3}$$

where $p$ is the conditional distribution over source action chunks $\mathbf{A}^{\text{src}}$, and $\mathbf{A}_t^{\text{src}}$ is a sampled initialization. The action expert $D^{(K)}$ refines this sample in $K$ denoising steps into the predicted executable chunk $\widehat{\mathbf{A}}_t$, conditioned on $\mathbf{h}_t$, $\mathbf{s}_t$, and $\mathbf{c}_t$, which summarizes recently committed actions. The motion-aligned source is intended to reduce the denoising steps required for each action chunk, while asynchronous execution reduces inference-induced waiting time and improves action smoothness across successive chunks.

### 3.2 Motion-Induced Latent Action from Historical Optical Flow

**Spatiotemporal flow representation.** Optical flow from the fixed head camera captures target and robot image-plane motion. A shared encoder $\Phi$ maps each flow field to $N$ spatial tokens with

Table 1: Performance comparison across different manipulation tasks. SR denotes success rate (%), and MS denotes the corresponding execution metric.

| Method | Adjust Bottle | | Beat Block Hammer | | Shake Bottle Horizontally | | Handover Mic | | Put Bottles Dustbin | | AVG | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SR↑ | MS↑ | SR↑ | MS↑ | SR↑ | MS↑ | SR↑ | MS↑ | SR↑ | MS↑ | SR↑ | MS↑ |
| OpenVLA | 0% | 1.73 | 0% | 4.23 | 6% | 12.33 | 0% | 2.34 | 0% | 7.25 | 1.2% | 5.58 |
| $\pi_0$ | 24% | 30.11 | 6% | 21.06 | 32% | 49.12 | 18% | 30.53 | 34% | 45.50 | 22.8% | 35.26 |
| $\pi_{0.5}$ | 10% | 13.85 | 4% | 15.21 | **56%** | **63.51** | 22% | 33.53 | 26% | 39.17 | 23.6% | 33.06 |
| PUMA | 24% | 27.73 | 22% | 33.36 | 16% | 43.58 | 20% | 44.90 | 28% | 49.77 | 22.0% | 39.87 |
| VLASH | 36% | 44.17 | 12% | 25.37 | 22% | 41.96 | 30% | 44.20 | 34% | 44.74 | 26.8% | 40.09 |
| **Ours** | **60%** | **62.02** | **26%** | **37.28** | 30% | 43.81 | **54%** | **65.30** | **38%** | **51.42** | **41.6%** | **51.97** |

additive spatial and temporal embeddings:

$$\mathbf{x}_{j,n} = \Phi\big(F^{\text{head}}_{j\to j+1}\big)_n + \mathbf{e}^{\text{pos}}_n + \mathbf{e}^{\text{time}}_{j-t} \tag{4}$$

where $j = t-L, \dots, t-1$ indexes flow intervals, $n = 1, \dots, N$ indexes spatial tokens, and $d$ is the feature dimension. The embeddings $\mathbf{e}^{\text{pos}}_n, \mathbf{e}^{\text{time}}_{j-t} \in \mathbb{R}^d$ encode image location and relative temporal position, respectively, with $j - t = -1$ denoting the latest interval. Stacking all $\mathbf{x}_{j,n} \in \mathbb{R}^d$ yields $\mathbf{X}_t \in \mathbb{R}^{LN\times d}$, preserving spatial layout and temporal order.

**Task-relevant motion aggregation.** A task-conditioned object query $\mathbf{q}^{\text{obj}}_t = Q(\mathbf{h}_t)$ and a learnable robot query $\mathbf{q}^{\text{robot}}$ attend to the shared flow tokens:

$$\mathbf{z}^{\text{obj}}_t = \text{Attn}\Big(\mathbf{q}^{\text{obj}}_t, \mathbf{X}_t, \mathbf{X}_t\Big), \qquad \mathbf{z}^{\text{robot}}_t = \text{Attn}\big(\mathbf{q}^{\text{robot}}, \mathbf{X}_t, \mathbf{X}_t\big) \tag{5}$$

where Attn denotes cross-attention with learned key and value projections. A learned relation module fuses the readouts as $\mathbf{z}^{\text{motion}}_t = R([\mathbf{z}^{\text{obj}}_t; \mathbf{z}^{\text{robot}}_t])$, where $[\,\cdot\,;\,\cdot\,]$ denotes concatenation. These operations implement $E$ in Equation 2, learning a feature-space relation without explicit physical relative-velocity estimation or three-dimensional motion decomposition. The mapper $G$ then combines this representation with $\mathbf{s}_t$ to obtain the configuration-dependent latent action $\mathbf{z}^{\text{act}}_t$. Together with $\mathbf{s}_t$, this coarse response conditions the source distribution in Section 3.3, while the action expert determines the precise executable trajectory under task and embodiment constraints.

### 3.3 Motion-Aligned Source Distribution for Few-Step Action Generation

**Conditional source parameterization.** We replace the motion-agnostic standard Gaussian with a conditional Gaussian in action space. For vectorized action dimension $d_A = Hd_a$, the source mapper predicts $(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) = P(\mathbf{z}^{\text{act}}_t, \mathbf{s}_t)$, where $\boldsymbol{\mu}_t \in \mathbb{R}^{d_A}$ is the mean response and $\boldsymbol{\Sigma}_t \in \mathbb{R}^{d_A\times d_A}$ is a positive-definite covariance controlling the scale and structure of variation. A motion-informed initialization is sampled as

$$\text{vec}(\mathbf{A}^{\text{src}}_t) \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \tag{6}$$

where vec denotes vectorization. This defines $p$ in Equation 3.

**Source learning and conditional denoising.** Given a demonstrated chunk $\mathbf{A}^\star_t$, we supervise the source and upstream motion-to-action mapping using Gaussian negative log-likelihood:

$$\mathcal{L}_{\text{src}} = -\log \mathcal{N}(\text{vec}(\mathbf{A}^\star_t); \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \tag{7}$$

To match the source used at inference, we train the expert with the conditional corruption

$$\mathbf{A}_{t,\gamma} = \alpha_\gamma \mathbf{A}^\star_t + \sigma_\gamma\, \text{sg}(\mathbf{A}^{\text{src}}_t) \tag{8}$$

where $\gamma \in [0,1]$ is diffusion time, and $\alpha_\gamma, \sigma_\gamma$ weight the clean action and source, respectively, with $(\alpha_0, \sigma_0) = (1, 0)$, $(\alpha_1, \sigma_1) = (0, 1)$, and $\sigma_\gamma > 0$ for $\gamma > 0$. The stop-gradient operator sg prevents denoising gradients from updating the source. With uniformly sampled $\gamma$, the expert learns clean-action prediction:

$$\mathcal{L}_{\text{den}} = \mathbb{E}_{\gamma, \mathbf{A}^{\text{src}}_t}\Big[\|D(\mathbf{A}_{t,\gamma}, \gamma \mid \mathbf{h}_t, \mathbf{s}_t, \mathbf{c}_t) - \mathbf{A}^\star_t\|^2_F\Big] \tag{9}$$

We optimize $\mathcal{L}_{\text{den}} + \lambda_{\text{src}}\mathcal{L}_{\text{src}}$, where $\lambda_{\text{src}} > 0$ weights source supervision.

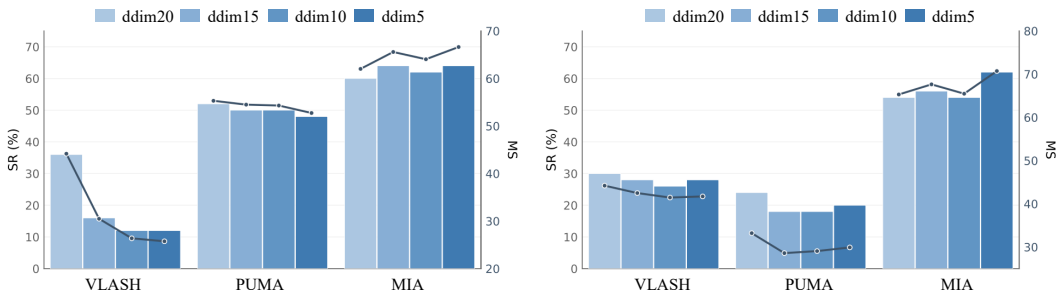



Figure 3: **Effect of DDIM denoising steps.** Success Rate (bars) and Manipulation Score (lines) on *adjust_bottle* (left) and *handover_mic* (right). Our method maintains strong performance with as few as 5 denoising steps, showing substantially lower sensitivity to sampling depth than the baselines.

**Few-step action generation.** At inference, we initialize $\mathbf{A}_{t,\gamma_K} = \mathbf{A}_t^{\text{src}}$ on the grid $1 = \gamma_K > \cdots > \gamma_0 = 0$ and apply deterministic DDIM updates (Song et al., 2021):

$$\begin{aligned} \widehat{\mathbf{A}}^{\star}_{t,k} &= D(\mathbf{A}_{t,\gamma_k}, \gamma_k \mid \mathbf{h}_t, \mathbf{s}_t, \mathbf{c}_t), \\ \mathbf{A}_{t,\gamma_{k-1}} &= \alpha_{\gamma_{k-1}} \widehat{\mathbf{A}}^{\star}_{t,k} + \frac{\sigma_{\gamma_{k-1}}}{\sigma_{\gamma_k}} \left( \mathbf{A}_{t,\gamma_k} - \alpha_{\gamma_k} \widehat{\mathbf{A}}^{\star}_{t,k} \right) \end{aligned} \tag{10}$$

Here, $\widehat{\mathbf{A}}^{\star}_{t,k}$ denotes the clean-action estimate at step $k$. The output $\widehat{\mathbf{A}}_t = \mathbf{A}_{t,\gamma_0}$ after $K$ updates implements $D^{(K)}$ in Equation 3. Motion enters through source construction, whereas $(\mathbf{h}_t, \mathbf{s}_t, \mathbf{c}_t)$ condition each refinement step.

### 3.4 Asynchronous Action Generation and Temporal Consistency

The robot executes a committed prefix of its action buffer while the policy generates new action from the latest observation. The new chunk begins at the prefix's handoff once generation is complete; committed commands remain unchanged, making the transition context available during inference.

**Overlapping inference and execution.** Let $\tau_t^{\text{gen}}$ denote the time required for observation processing, source construction, and denoising, and $\tau_t^{\text{rem}}$ the remaining prefix duration at inference launch. The handoff waiting time is

$$\tau_t^{\text{wait}} = \max(0, \tau_t^{\text{gen}} - \tau_t^{\text{rem}}) \,. \tag{11}$$

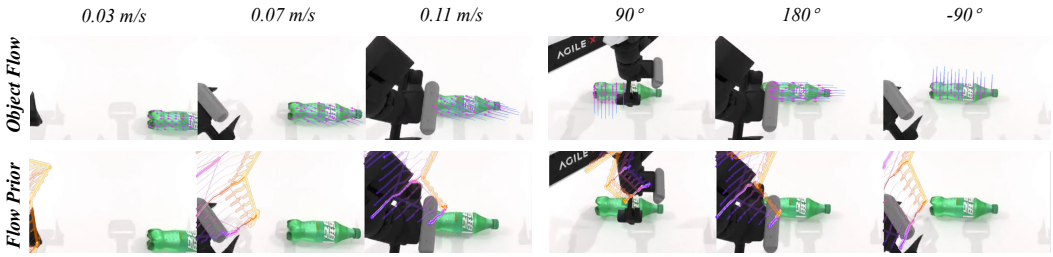
Inference is fully overlapped when it finishes within the remaining execution window. Few-step refinement shortens generation, while asynchronous scheduling reduces the portion exposed as a control pause.

**Conditioning on execution context.** To account for the robot motion preceding the handoff, we condition the action expert on the most recent $M$ commands committed to execution:

$$\mathbf{c}_t = C_\theta([\mathbf{b}_{t,1}, \ldots, \mathbf{b}_{t,M}], \boldsymbol{\delta}_t) \,. \tag{12}$$

Here, $\mathbf{b}_{t,m} \in \mathbb{R}^{d_a}$ is the $m$-th command in temporal order, and $\boldsymbol{\delta}_t \in \mathbb{R}^M$ records execution-time offsets relative to observation time $t$, distinguishing executed commands from those scheduled during inference. The temporal encoder $C_\theta$ maps these commands and their timing to $\mathbf{c}_t$ and is trained with the action expert. Each target chunk is paired with preceding commands and an observation window aligned to the same handoff convention. This context enters the denoising objective (Equation 9), conditioning the leading actions on preceding motion to promote smooth transitions while allowing adaptation to updated observations.

Recent action context also provides a potential route for carrying information from earlier motion-conditioned predictions across updates. We examine whether this supports object permanence during temporary target occlusion in Section 4.




(a) Effect of object speed on flow (b) Effect of object direction on flow

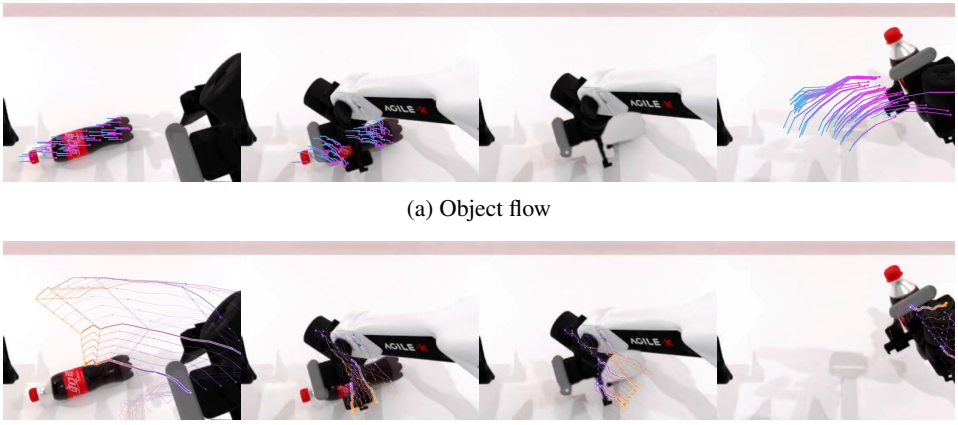
Figure 4: **Flow-prior response to object motion dynamics.**The left panel illustrates the effect of object speed under a fixed motion direction. The right panel illustrates the effect of object motion direction at a fixed speed. Variations in object speed and motion direction are clearly reflected in the predicted flow prior, providing motion-conditioned guidance for robot manipulation.

# 4 EXPERIMENTS

## 4.1 EXPERIMENT SETUP

**Simulation Setup.** We evaluate our method on five representative dynamic manipulation tasks selected from the DOMINO benchmark: `adjust_bottle`, `handover_mic`, `beat_hammer_bell`, `shake_bottle_horizontally`, and `put_bottles_dustbin`. For each task, we fine-tune the model using 50 demonstrations and evaluate it on 50 test episodes. The evaluation instructions are unseen during training, and a fixed evaluation seed is used to ensure reproducibility. We report two metrics: Success Rate (SR) and Manipulation Score (MS).

**Real-world Setup.** We use a RealMan RX75 robotic arm and collect 150 demonstrations in each of two settings: Conveyor (Slow) and standard Rolling Bottle without distractors. All policies are trained on these same two demonstration sets. Conveyor (Fast), Conveyor (Occluded), and Rolling Bottle (SR), which introduces a randomly rolling distractor, are evaluated zero-shot without condition-specific demonstrations or further fine-tuning. SR in the task name denotes spatial reasoning. Each policy is evaluated on 20 rollouts per condition.



(a) Object flow

(b) Flow prior response

Figure 5: **Flow prior under temporarily masked object motion.** We temporarily mask the bottle-associated flow during motion. Even without the instantaneous object-motion cue, the predicted flow prior preserves the overall motion trend, demonstrating its ability to infer persistent dynamics from temporal context.

## 4.2 COMPARISON WITH STATE-OF-ART MODELS

As shown in Table 1, our method achieves the strongest overall performance across five dynamic manipulation tasks, with an average SR of 41.6% with a sizable 14.8% improvement over previous

methods, setting a new state-of-the-art in dynamic manipulation. The advantage is most pronounced on tasks that require continuous adaptation to object motion. For example, on *handover_mic*, our method reaches 54% SR, substantially outperforming the strongest baseline by 24%. Consistent gains are also observed on object adjustment, tool interaction, and goal-directed placement. Overall, these results suggest that explicitly modeling relative motion improves the policy's ability to maintain effective interaction under changing object dynamics, rather than merely improving performance on a specific task.

### 4.3 Ablation Study

We study how Historical Flow (HF), Flow Prior (FP), and asynchronous execution contribute to the final performance in Table 2. A notable observation is that simply introducing historical flow as additional visual input does not improve the policy (SR: 26% $vs.$ 15%). This suggests that additional temporal information is not automatically useful unless the policy can properly interpret and act on it.

Table 2: Ablation of the proposed components.

| | Components | | | Average | |
|---|---|---|---|---|---|
| **Set** | **HF** | **FP** | **Async.** | **SR** | **MS** |
| 1 | × | × | × | 26% | 39.00 |
| 2 | ✓ | × | × | 15% | 29.31 |
| 3 | ✓ | ✓ | × | 28% | 39.80 |
| 4 | ✓ | × | ✓ | 27% | 40.09 |
| 5 | ✓ | ✓ | ✓ | **42%** | **51.97** |

The performance raises when HF is paired with either FP (SR: 15% $vs.$ 28%) or asynchronous execution (SR: 15% $vs.$ 27%). Both variants recover the lost performance, showing that motion history becomes useful once it is transformed into predictive motion cues or aligned with the future execution state. Most importantly, enabling all three components yields the strongest result, reaching 42% SR and 51.97 MS.

Rather than contributing independently, the three components therefore appear to form a complementary pipeline: HF provides temporal context, FP converts this context into an anticipation of future motion, and asynchronous execution allows the policy to act on that anticipation with reduced perception–action mismatch. This interaction explains why the full model improves substantially more than any partial configuration.

### 4.4 Qualitative Analysis

We next examine what information is captured by the learned flow prior and how it affects action generation beyond the final task success rate.

**Efficient action generation.** We first examine how performance changes as the number of DDIM denoising steps is reduced. As shown in Fig. 3, our method remains stable even with only five sampling steps on both *adjust_bottle* and *handover_mic*, whereas the baselines become increasingly sensitive to aggressive step reduction. This robustness suggests that the learned motion prior constrains action generation toward dynamically plausible solutions, reducing the need for extensive iterative refinement. In dynamic manipulation, this property is particularly useful because fewer denoising steps directly translate into lower inference latency and more frequent replanning.

**Motion-conditioned responses.** We then vary object velocity and direction independently to probe the behavior encoded by the flow prior. As shown in Fig. 4, when object speed increases, the predicted robot motion becomes more anticipatory and spatially extended, consistent with the larger displacement that must be compensated during execution. Changing the motion direction produces qualitatively different strategies, including upper-side interception, left-side tracking, and diagonal tracking. These responses show that the prior does not simply reproduce optical flow, but translates object dynamics into interaction-dependent robot motion.

**Object permanence.** Finally, we temporarily mask the object-associated optical flow to test whether the learned prior depends only on instantaneous motion observations. As shown in Fig. 5, the predicted robot prior continues to preserve the overall motion tendency even when the object-flow cue briefly disappears. This indicates that the model integrates temporal context to maintain an internal expectation of the ongoing dynamics, rather than recomputing its response solely from the current optical flow.

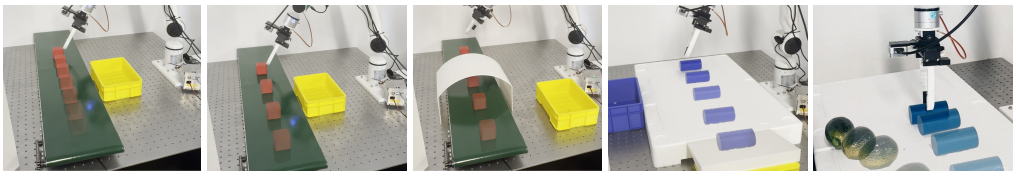



Figure 6: **Real-world dynamic manipulation tasks.** Grasping cubes from a conveyor under slow, fast, and occluded conditions, and grasping a rolling bottle with and without a randomly rolling distractor. SR denotes spatial reasoning.

### 4.5 Real-World Experiments

We test whether motion-guided action generation supports capabilities beyond the demonstrated conditions across two task families (Fig. 6), with each task family using only one kind of demonstrations (described in 4.1). Table 3 shows that MIA achieves 88% average success, compared with 66% for PUMA and 71% for VLASH. On the three conditions absent from the demonstrations—Conveyor (Fast), Conveyor (Occluded), and Rolling Bottle (SR)—its average success is 86.7%, versus 55.0% and 63.3% for PUMA and VLASH, respectively.

Table 3: Real-world success rates (%) over 20 rollouts per condition. All policies use the same demonstrations from Conveyor (Slow) and standard Rolling Bottle; the other three conditions test zero-shot transfer.

| **Policy** | Conv. (Slow) | Conv. (Fast) | Conv. (Occl.) | Rolling Bottle | Rolling Bottle (SR) | **Avg.** |
|---|---|---|---|---|---|---|
| PUMA | 85 | 55 | 50 | 80 | 60 | 66 |
| VLASH | 85 | 60 | 75 | 80 | 55 | 71 |
| **MIA (ours)** | **95** | **85** | **90** | **85** | **85** | **88** |

**Adaptation to faster motion.** With conveyor demonstrations limited to the slow setting, MIA achieves 85% success in Conveyor (Fast). This supports transfer across target speeds, consistent with an action prior that changes with observed motion magnitude (Fig. 4 (a)).

**Robustness to temporary occlusion.** PUMA shows the largest drop under unseen occlusion, from 85% to 50%, versus 95% to 90% for MIA and 85% to 75% for VLASH. PUMA's world-model-style supervision uses auxiliary prediction of future object features during training (Fang et al., 2026). Its larger drop suggests that this predictive representation may struggle to capture object motion through unseen occlusions. MIA directly uses observed motion history to guide action generation, supporting continued manipulation during temporary visual interruptions (Fig. 5).

**Target selection amid moving distractors.** Rolling Bottle (SR) adds an independently rolling distractor to the standard setting. MIA maintains 85% success in both conditions, whereas PUMA drops from 80% to 60% and VLASH from 80% to 55%. The distractor introduces additional scene motion without changing the target or grasping objective. The absence of an observed success-rate drop supports selective use of task-relevant motion, consistent with task-conditioned motion aggregation guiding the action prior toward the target despite unrelated movement.

## 5 Conclusion

We presented MIA, a framework for dynamic manipulation that combines motion-informed diffusion initialization with context-aware asynchronous execution. By mapping task-relevant motion history to the action source distribution, MIA directly couples observed target dynamics with action generation, without explicitly forecasting future scenes. Simulation experiments demonstrate improved overall performance and effective five-step inference. Real-world evaluations further show transfer from two demonstrated settings to faster motion, temporary occlusion, and moving distractors within the same task families, without condition-specific demonstrations or additional fine-tuning. Together, these findings support motion-conditioned action priors as an effective inductive bias for generalization beyond demonstrated motion and observation conditions.

### AI USE STATEMENT

In this work, we used generative AI tools to polish writing. We have not used generative AI tools for generating synthetic datasets and auto research. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## REFERENCES

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, brian ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 27–30 Sep 2025.

Kevin Black, Manuel Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. *Advances in Neural Information Processing Systems*, 38:33383–33407, 2026.

Wenxi Chen, Kaidi Zhang, Chi Lin, Zhiyuan Zhang, Yu She, Yuejiang Liu, Raymond A Yeh, Shaoshuai Mou, and Yan Gu. Dream-chunk: Reactive action chunking with latent world model. *arXiv preprint arXiv:2606.18589*, 2026.

Zhuoqun Chen, Xiu Yuan, Tongzhou Mu, and Hao Su. Responsive noise-relaying diffusion policy: Responsive and efficient visuomotor control. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

Meipo Dai, Qiyuan Zhuang, He-Yang Xu, Ying-Jie Shuai, Yijun Wang, Qi Dou, and Xiu-Shen Wei. Where should action generation begin? a learnable source prior for generative robot policies. *arXiv preprint arXiv:2606.17408*, 2026.

Heng Fang, Shangru Li, Shuhan Wang, Xuanyang Xi, Dingkang Liang, and Xiang Bai. Towards generalizable robotic manipulation in dynamic environments. In *European Conference on Computer Vision (ECCV)*, 2026.

Taisei Hanyu, Nhat Chung, Huy Le, Toan Nguyen, Yuki Ikebe, Anthony Gunderman, Ho Minh Duy Nguyen, Khoa Vo, Tung Kieu, Kashu Yamazaki, Chase Rainwater, Anh Nguyen, and Ngan Le. Slotvla: Towards modeling of object-relation representations in robotic manipulation. In *Proceedings of ICRA 2026. IEEE International Conference on Robotics and Automation (ICRA-2026)*. ICRA, 2026.

Sigmund H Høeg, Yilun Du, and Olav Egeland. Fast policy synthesis with variable noise diffusion models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4821–4828. IEEE, 2025.

Motonari Kambara, Koki Seno, Tomoya Kaichi, Yanan Wang, and Komei Sugiura. Lilac: Language-conditioned object-centric optical flow for open-loop trajectory generation. *IEEE Robotics and Automation Letters*, 11(6):6767–6774, 2026.

Sinjae Kang, Chanyoung Kim, Kaixin Wang, Li Zhao, and Kimin Lee. Warmprior: Straightening flow-matching policies with temporal priors. *arXiv preprint arXiv:2605.13959*, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 06–09 Nov 2025.

Yunfan Lou, Hewen Gao, Xiyu Zhu, Zhuoran Qiao, Xuan Han, Yifan Yang, Yifan Ye, Boxian Yao, and Zhibo Pang. Dynamicwam: Dual-path motion conditioning for world-action models in dynamic manipulation. *arXiv preprint arXiv:2608.00793*, 2026.

Yuxiang Lu, Zhe Liu, Xianzhe Fan, Zhenya Yang, Jinghua Hou, Junyi Li, Kaixin Ding, and Hengshuang Zhao. Faster: Rethinking real-time flow vlas. *arXiv preprint arXiv:2603.19199*, 2026.

Yunchao Ma, Yizhuang Zhou, Yunhuan Yang, Tiancai Wang, and Haoqiang Fan. Running vlas at real-time speed. *arXiv preprint arXiv:2510.26742*, 2025.

Bruno Machado, Alexandre Chapin, Emmanuel Dellandrea, and Liming Chen. Flowing with purpose: Latent action guided flow matching policies for robotic manipulation. *arXiv preprint arXiv:2606.23420*, 2026.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Shahram Najam Syed, Arthur Jakobsson, Haoran Hao, and Jeffrey Ichnowski. Intercepting the future: Latent-space predictive world model for dynamic vla manipulation. *arXiv preprint arXiv:2606.02486*, 2026.

Jiaming Tang, Yufei Sun, Yilong Zhao, Shang Yang, Yujun Lin, Zhuoyang Zhang, James Hou, Yao Lu, Zhijian Liu, and Song Han. Vlash: Real-time vlas via future-state-aware asynchronous inference. *arXiv preprint arXiv:2512.01031*, 2025.

Claes von Hofsten. Catching skills in infancy. *Journal of Experimental Psychology: Human Perception and Performance*, 9(1):75, 1983.

Kuanning Wang, Ke Fan, Chenhao Qiu, Zeyu Shangguan, Yuqian Fu, Yanwei Fu, Daniel Seita, and Xiangyang Xue. Oflow: Injecting object-aware temporal flow matching for robust robotic manipulation. *arXiv preprint arXiv:2604.17876*, 2026a.

Xinkai Wang, Chenyi Wang, Yifu Xu, Mingzhe Ye, Fucheng Zhang, Jialin Tian, Xinyu Zhan, Lifeng Zhu, Cewu Lu, and Lixin Yang. Lamp: Learning vision-language-action policy with 3d scene flow as la tent m otion p rior. In *European Conference on Computer Vision*, pp. 227–244. Springer, 2026b.

William H Warren. The dynamics of perception and action. *Psychological review*, 113(2):358, 2006.

Haozhe Xie, Beichen Wen, Jiarui Zheng, Zhaoxi Chen, Fangzhou Hong, Haiwen Diao, and Ziwei Liu. Dynamicvla: A vision-language-action model for dynamic object manipulation. *arXiv preprint arXiv:2601.22153*, 2026.

Yanyan Zhang, Chaoda Song, Vikash Singh, Xinpeng Li, Kai Ye, Zhe Hu, Zhongzhu Pu, Yu Yin, and Vipin Chaudhary. Overcoming dynamics-blindness: Training-free pace-and-path correction for vla models. *arXiv preprint arXiv:2605.11459*, 2026.

Yifan Zhang, Ruiping Wang, and Xilin Chen. Dynamic behavior cloning with temporal feature prediction: Enhancing robotic arm manipulation in moving object tasks. *IEEE Robotics and Automation Letters*, 10(6):5209–5216, 2025.