ROBOTIC SPATIOTEMPORAL EDITING: A UNIFIED INTERFACE FOR CONTROLLABLE ACTION GENERATION VIA MASKED ACTION RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative policies can synthesize robot behaviors, yet controlling and editing them through diverse spatiotemporal references remains challenging. Real-world manipulation requires robots to satisfy user-specified constraints on where, when, and how actions are executed. Our key insight is that spatiotemporal control references, such as keypoints, flows, orientations, and keyposes, can be viewed as partial specifications of the same temporal Cartesian action space. We introduce the **MAP (Masked Action Space)** interface to unify these action references and formulate controllable manipulation as an action inpainting problem. Our **Inpainting Policy (IP)** reconstructs missing action dimensions of the end-effector, combining progress-guided alignment with multi-mask training to support unseen action constraint combinations. We further adopt a training-free **RePaint** variant that iteratively inpaints the action during denoising to reduce control error. On ManiSkill and LIBERO-10, our policy achieves average task success rates of 92.67% and 64.67%, respectively. Real-world experiments demonstrate spatiotemporal control with average positional and rotational errors of $44.3\,\mathrm{mm}$ and $7.8°$, respectively. Videos of reference-guided robot manipulation are available at `https://masked-am.github.io/Inpainting-Policy-Website/`.

1 INTRODUCTION

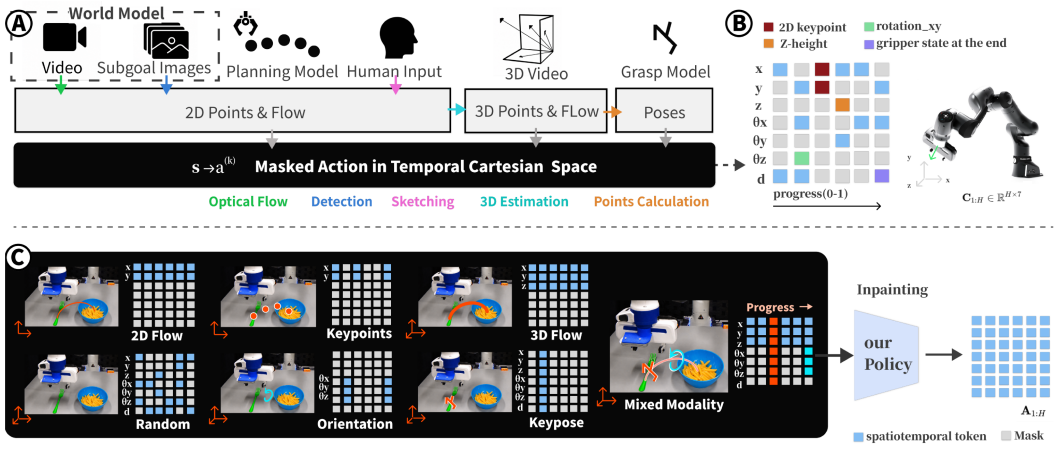


Figure 1: **Unified spatiotemporal control via action inpainting. Top:** The **Masked Action Space (MAP)** interface unifies heterogeneous references from upstream models and human inputs as partial specifications of temporal Cartesian actions. We use progress values instead of raw **timesteps**. **Bottom:** Our **Inpainting Policy (IP)** completes these partial specifications through action inpainting, supporting potentially unlimited spatiotemporal reference types and unseen combinations.

Generative models have recently demonstrated remarkable progress in robotic manipulation, enabling policies to synthesize complex robot behaviors from visual observations and task descrip-

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.

tions (Chi et al., 2025; Black et al., 2026; Brohan et al., 2023). However, generating plausible behaviors alone is insufficient for deploying robots in the physical world. Real-world tasks require robots not only to decide what to do but also to satisfy constraints on **where, when, and how** to execute an action (Rempe et al., 2026). Moreover, humans and intelligent systems often need to refine, modify, and compose robot behaviors by adding spatiotemporal constraints. For example, when tidying a table, a user may use a hand gesture or a sketch instruction (Li et al., 2025b) to specify where an object should be placed, rather than merely asking the robot to tidy the table. While generative models can create robot behaviors, the next challenge is enabling humans and intelligent systems to control and edit them.

Meanwhile, robotics is gaining access to diverse spatiotemporal references from world models, 3D/4D videos, advanced agents, grasp models, and human demonstrations (Jang et al., 2025; Gao et al., 2026; Chen et al., 2026b; Murali et al., 2025; Fan et al., 2025b), including keypoints (Huang et al., 2024), flow (Xu et al., 2024), keyposes (Yu et al., 2024), and grasp proposals (Fang et al., 2023; Hao et al., 2025) (see Figure 1(A)). These representations are physically grounded and low-dimensional, but they only provide partial information on future robot motion rather than complete executable actions. Existing approaches typically address this gap with signal-specific policies or conditioning mechanisms. This motivates the need for **a unified action interface** that can transform diverse spatiotemporal references into controllable robot behaviors.

Our key insight is that heterogeneous spatiotemporal control are not different control problems, but partial action tokens of the same underlying Temporal Cartesian Action Space. We design **Masked Action Space (MAP)** as a shared interface that represents these references through partially specified action tokens and a binary mask (see Figure 1(B,C)). We formulate controllable action generation as **a masked action inpainting problem**, and we propose **Inpainting Policy (IP)** to reconstruct masked components in MAP and turn them into a complete executable trajectory (see Section 3.3).

To enable this formulation, we design three components. **(1)** A **progress-guided sliding window** aligns control references with the current task stage using estimated progress value (Chen et al., 2026a) rather than raw execution timesteps (Section 3.2). **(2)** A **two-stage joint-training strategy** uses random masks in the first stage and structured masks in the second stage to learn robust control over diverse and unseen control references (Section 3.4). **(3)** As a **training-free** alternative, we also adopt a **RePaint-style inpainting strategy** that can be applied during the inference-time denoising process of an off-the-shelf policy. We conduct experiments on 15 tasks across the ManiSkill, LIBERO-10, and real-world environments, covering six standard control references and randomly structured references. In sum, our method demonstrates several capabilities:

- **Versatile controllability.** Inpainting Policy supports diverse spatiotemporal references, including keypoints, flows, orientations, poses, and combinations of these reference types. In real-world experiments, it attains mean position and rotation errors of $44.3\,\mathrm{mm}$ and 7.8°, respectively. Meanwhile, our method improves the success rate on LIBERO-10 by 4.67 percentage points compared with the baseline. This proves that our policy enables arbitrary spatiotemporal controllability without compromising task completion.
- **Interpretability.** MAP (Masked Action Space) provides a human-readable and editable interface that bridges multi-dimensional spatiotemporal information from the physical world with robotic manipulation actions via the masked temporal Cartesian space. It explicitly indicates which timesteps and action dimensions are constrained.
- **Practicality.** Our method can be directly applied to diffusion- or flow matching-based policies in simulated or real-world environments. Built on a lightweight $49\mathrm{M}$-parameter DiT model, it achieves a 92.67% success rate and a control error of 0.023 (Table 1) in ManiSkill experiments. In real-world experiments, a $118\mathrm{M}$-parameter diffusion model achieves a success rate above 90%.

2 Related Work

2.1 Heterogeneous Spatiotemporal Signals for Robot Control

Robotic manipulation increasingly uses spatially grounded intermediate representations, including keypoints, trajectories, and keyposes, to guide action generation. Point-based methods provide

sparse spatial guidance (Huang et al., 2024; Liu et al., 2024), while KALM uses task-relevant keypoints to define object-relative frames for action prediction (Fang et al., 2025). BiKC predicts keyposes to guide a low-level trajectory generator, and AnyGrasp provides grasp-pose proposals for downstream execution (Yu et al., 2024; Fang et al., 2023). Gemini Robotics-ER supports spatial reasoning and trajectory prediction, while RoboBrain and A0 produce affordance and trajectory predictions to guide manipulation (Team et al., 2025; Ji et al., 2025; Xu et al., 2026). Beyond sparse references, trajectory-based interfaces provide denser motion guidance for downstream action policies (Xu et al., 2024; Wen et al., 2024; Gu et al., 2023; Liu et al., 2025). Video-based models offer another route to control: V-JEPA 2-AC plans toward image goals through action-conditioned prediction in latent space, while UVA jointly learns video and action prediction (Assran et al., 2025; Li et al., 2025a).

Our work focuses on **unifying heterogeneous control references** as specified entries in a shared Cartesian action representation, allowing a single policy to reconstruct the remaining action entries across different reference types and their combinations.

2.2 Inpainting for Robot Control

Visual masking and inpainting support several aspects of robot learning. ROSIE augments robot observations through text-guided inpainting, while Mirage modifies robot appearance through cross-painting for embodiment transfer (Yu et al., 2023; Chen et al., 2024). Cloak masks the end-effector in wrist-camera observations without generative inpainting, whereas Masked Visual Actions uses partially revealed video trajectories for forward and inverse world modeling (Piseno et al., 2026; Alzayer et al., 2026). In motion generation, Kimodo supports pose-feature completion, Scheduled Inpainting balances motion preservation and editing, and MotionBricks performs latent in-betweening (Rempe et al., 2026; Agrawal et al., 2026; Wang et al., 2026b). In action space, RTC and DiscreteRTC preserve committed action prefixes while generating subsequent actions, enabling asynchronous execution through inpainting (Black et al., 2025; Wang et al., 2026a). HDP and DISCO use inpainting to impose trajectory-endpoint and semantic-keyframe constraints, respectively (Ma et al., 2024; Hao et al., 2025).

3 Method

Our method first introduces a unified controllable interface, the **MAP (Masked Action Space)**, for manipulation action generation, which converts heterogeneous spatiotemporal signals into specified action tokens and associated binary masks, thereby representing different control references in a common masked Cartesian action space (Section 3.1). Based on this representation, we develop an **Inpainting Policy** that reconstructs the masked action tokens and generates executable action chunks (Section 3.3). Our framework consists of three components. **(1)** A progress-guided sliding window aligns MAP with the current task progress and provides a temporally anchored MAP condition (Section 3.2). **(2)** Multi-modal joint training over heterogeneous mask families enables structured control and zero-shot generalization to unseen spatiotemporal compositions (Section 3.4). **(3)** We adopt RePaint (Lugmayr et al., 2022) as a training-free inference strategy, which repeatedly reinserts the specified action tokens during the denoising process of the diffusion-based policy, thereby improving adherence to the provided constraints (Section 3.5).

3.1 Masked Action Space

We formulate reference-guided robotic manipulation as reconstructing complete actions from partial specifications in a temporal Cartesian action space.

We use normalized task progress $p' \in [0, 1]$ as the temporal coordinate, with reference alignment detailed in Section 3.2. Let $A^{\text{ref}} \in \mathbb{R}^{T_{\text{ref}} \times 7}$ denote the reference action-value array at discrete progress locations p'_p, where $p = 0, \ldots, T_{\text{ref}} - 1$. At each location, the action vector is defined as $a_p = (x_p, y_p, z_p, \theta_{x,p}, \theta_{y,p}, \theta_{z,p}, g_p)$, where the first six dimensions describe the gripper pose and g_p denotes the gripper state. Each scalar action entry is a spatiotemporal action token. The spatiotemporal references are represented by a binary mask $M \in \{0, 1\}^{T_{\text{ref}} \times 7}$ and the corresponding specified action tokens $M \odot A^{\text{ref}}$, where $M_{p,d} = 1$ indicates a supplied value and $M_{p,d} = 0$ indicates an unspecified entry.

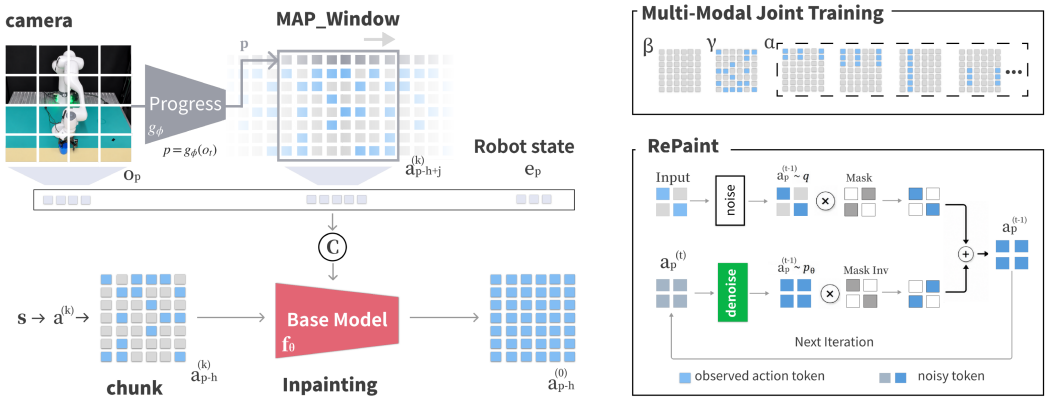


Figure 2: **Inpainting Policy**. Given camera observations and proprioceptive information, a progress network anchors a local MAP window and masked action chunk in the temporal Cartesian action space. The window is encoded into a feature vector to condition the base model, which predicts missing action tokens from the masked action chunk. Multi-modal joint training enables the policy to generalize to unseen spatiotemporal modalities.

Mapping Control References to Masks. As illustrated in Figure 1(C), different distributions of M instantiate different control signals. These references can be continuous or discrete along the temporal axis. Structured mask types include 2D/3D keypoints, 2D/3D flow, keyposes, orientations, gripper states, and mixed-modality conditions formed by randomly combining these references, as illustrated in Figure 1. Here, 2D references specify Cartesian x, y coordinates and flow references denote position trajectories; image-space references must first be mapped into the action coordinate system. For a reference at progress p', let $p^\star = \arg\min_p |p'_p - p'|$ denote its discrete reference index. For example, a 3D keypoint activates $M_{p^\star,\{1,2,3\}} = 1$, while an orientation reference activates $M_{p^\star,\{4,5,6\}} = 1$. MAP also supports unstructured mask types, as shown in Figure 1(B), such as specifying the gripper state at a particular manipulation stage, constraining the end-effector state along the z-axis, or specifying a rotation angle on a particular plane.

3.2 Progress-Guided Sliding Window of MAP

We use normalized task progress $p' \in [0, 1]$ to align references with policy execution because execution speed can vary across rollouts, making raw timesteps unreliable for identifying corresponding task stages (see the related ablation study in Supplementary Section 3.5). Given the current observation $o_t = (I_t, s_t)$, where I_t denotes camera observations and s_t denotes the robot state, a progress network predicts $p'_t = g_\phi(o_t) \in [0, 1]$. The corresponding reference index is $p_t = \arg\min_p |p'_p - p'_t|$.

The progress estimate selects a local reference window $\mathcal{W}_t = [a_{p_t-h}, \ldots, a_{p_t}, \ldots, a_{p_t+j}]$, where h and j specify the past and future context lengths. The encoder receives only the specified entries and their visibility masks; boundary handling and window-length conventions are detailed in Supplementary Section 1.1. The window associates reference constraints with the current task stage and provides nearby reference context for action generation. Its temporal coverage is controlled by the backward and forward window lengths, whose effect is evaluated in Supplementary Section 3.2.

3.3 Inpainting Policy

As illustrated in Figure 2, IP generates complete Cartesian action chunks conditioned on visual observations, robot state, and the local MAP window selected in Section 3.2. A 2D convolutional encoder maps the local MAP window to a 128-dimensional feature vector, which is concatenated with visual and proprioceptive features to condition the denoising network (see Supplementary Sections 1.6 and 3.1 for architecture details and encoder comparisons). Let $A^{(0)} \in \mathbb{R}^{H\times 7}$ denote a clean action chunk and $A^{(k)} = \sqrt{\bar{\alpha}_k} A^{(0)} + \sqrt{1-\bar{\alpha}_k}\epsilon$ its noisy version at diffusion step k, where $\bar{\alpha}_k$ is the cumulative signal-retention coefficient and $\epsilon \sim \mathcal{N}(0, I)$. The denoising network predicts $\widehat{\epsilon} = \epsilon_\theta(A^{(k)}, k, C_t)$, where C_t contains the observation features and the encoding of $\mathcal{W}_t$. Let

$M_t^{\text{chunk}} \in \{0,1\}^{H\times 7}$ denote the reference visibility mask aligned with the target action chunk. The training objective is

$$\mathcal{L}_{\text{MAP}} = \mathbb{E}\left[w_m \left\|M_t^{\text{chunk}} \odot (\epsilon - \widehat{\epsilon})\right\|_F^2 + w_u \left\|(1 - M_t^{\text{chunk}}) \odot (\epsilon - \widehat{\epsilon})\right\|_F^2\right],$$

where w_m and w_u control the relative importance of known and unknown regions, and the expectation is over training examples, sampled masks, diffusion steps, and Gaussian noise. In practice, we use a weighted reconstruction objective to encourage consistency with specified references while learning to reconstruct unspecified actions.

3.4 Multimodal Joint Training

To enable one model to handle diverse spatiotemporal conditions, we construct a multi-modal masked action dataset from demonstrations. We train the Inpainting Policy with three complementary mask patterns: α, β, and γ, each targeting a distinct aspect of masked action reconstruction.

The α-pattern consists of structured spatiotemporal control signals, including 2D keypoints/flow, 3D keypoints/flow, keyposes, orientation, and gripper-state conditions. These masks represent common controllable action-generation interfaces, where partial Cartesian action tokens are specified as subgoals or constraints. Joint training over α enables the policy to follow diverse structured conditions within a unified action space.

The β-pattern corresponds to the no-known-context case, where no action reference is provided. In this setting, the model reduces to plain action generation by the diffusion policy. Including β encourages robustness when test-time conditions are sparse, such as low-frequency control signals.

The γ-pattern introduces random token-level masks. Since structured masks are special cases of random masking, γ promotes token-level generalization across MAP, especially under OOD patterns. However, experiments show that random-mask training may underperform structured joint training when training data is limited.

We sample a mask family $c \sim \rho(c)$, where $c \in \{\alpha, \beta, \gamma\}$, and then draw a mask $M \sim p(M \mid c)$. The distribution $\rho(c)$ controls the relative frequency of structured, no-reference, and random-mask examples during policy training. Mask families and compositions are detailed in Supplementary Sections 1.4–1.5. For LIBERO-10, we first train with random masks and then continue with a mixture of structured masks, as detailed in Supplementary Sections 2.2 and 3.4.

We find that, after joint training, the model can follow compositions of different spatiotemporal constraints and generaloize to an OOD mask pattern, formed by combining several standard patterns but never observed during training (Figure 5).

3.5 Inpainting via RePaint

We adopt a RePaint-style inference strategy (Lugmayr et al., 2022) that repeatedly reinserts specified reference values during denoising without additional policy training. Given clean reference values $R_t \in \mathbb{R}^{H\times 7}$ aligned with the target action chunk, we sample their noisy version through forward diffusion. The binary mask M_t^{chunk} specifies the observed action tokens, while $1 - M_t^{\text{chunk}}$ indicates the unknown tokens to be reconstructed.

At each reverse denoising step k, we obtain a denoised proposal $\widetilde{A}^{(k-1)} = \mathcal{D}_\theta(A^{(k)}, k, C_t)$, noise the reference values to the corresponding level as $R_t^{(k-1)} = \sqrt{\bar{\alpha}_{k-1}} R_t + \sqrt{1-\bar{\alpha}_{k-1}}\xi_k$, and update the action chunk by replacing the known entries:

$$A^{(k-1)} = M_t^{\text{chunk}} \odot R_t^{(k-1)} + (1 - M_t^{\text{chunk}}) \odot \widetilde{A}^{(k-1)},$$

where $\mathcal{D}_\theta$ denotes one reverse diffusion step using the noise prediction $\epsilon_\theta(A^{(k)}, k, C_t)$, $\xi_k \sim \mathcal{N}(0, I)$, and C_t denotes the conditioning features, including visual features, robot state, and the MAP window.

The final action chunk is obtained by overwriting the known tokens with clean reference values at the last denoising step ($\bar{\alpha}_0 = 1$). This overwrite operation is applied at every denoising step, ensuring that the final sampled action chunk strictly preserves the specified reference values rather than merely conditioning on them.

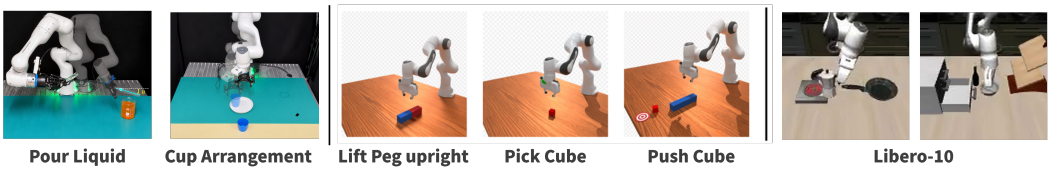


Figure 3: Simulation and real-world tasks.

4 EXPERIMENTS

Our experiments focus on three questions: (1) What information in physical references matters for action reconstruction? (2) Can the model generalize to unseen, out-of-distribution mask patterns? (3) How does each component contribute to the performance of our method? We evaluate our policy on 15 tasks across ManiSkill, LIBERO-10, and real-world environments, demonstrating diverse manipulation skills and long-horizon capabilities. Figure 3 provides an overview of the simulation and real-world tasks.

Dataset Construction. For the simulation experiments, we construct action references by masking demonstration trajectories, without collecting additional reference annotations. Pairing each demonstration with multiple mask configurations provides diverse reference conditions while preserving its observations and target actions. As shown in Figure 1(C), structured masks reveal modality-specific coordinate groups at selected reference locations for 2D/3D points, 7D poses, and orientations, or retain position coordinates along reference trajectories for 2D/3D flow. Random masks instead reveal individual scalar action entries across reference locations and action dimensions (Section 3.1). We additionally evaluate `Mix0`, a composition excluded from joint training that combines dense 2D flow, one 7D pose, and three 3D points (see Supplementary Section 2.4).

Progress Module. Our progress module adapts SARM (Chen et al., 2026a) to estimate task progress from frozen CLIP image and text features and proprioceptive states. The module is trained separately from the policy and uses an eight-layer Transformer with a hidden dimension of 768 and a temporal sampling interval of two frames.

Metrics. We report task success rate (SR) to evaluate task completion. In addition to SR, we introduce **Control Error (CE)** to quantify how closely the generated trajectory follows the prescribed control signals. Let $\tau = (a_1, a_2, \ldots, a_T)$ denote the generated action trajectory, where $a_t \in \mathbb{R}^7$, and let $\mathcal{C} = \{c_1, c_2, \ldots, c_K\}$ denote the control signals. For each signal c_ℓ, a selection matrix S_ℓ extracts the corresponding action dimensions. We define CE as the average distance from each control signal to its closest matching trajectory point: $\text{CE} = \frac{1}{K}\sum_{\ell=1}^{K} \min_{t\in\{1,\ldots,T\}} \|S_\ell \cdot a_t - c_\ell\|_2$. Lower CE indicates closer agreement with the prescribed control values.

Baseline. We compare our method with the following baselines. All methods are trained and evaluated using the same data, random seed, and initial states. **(1) Diffusion Policy (DP).** We use Diffusion Policy (Chi et al., 2025) with a U-Net backbone as the standard behavior cloning baseline, generating actions without control references. **(2) DiT Policy.** We replace the U-Net with a Diffusion Transformer (DiT) (Peebles & Xie, 2023) to isolate the effect of the backbone. **(3) Spatiotemporal Conditioning (STC).** Following prior work on trajectory-, keypose-, and motion-conditioned action generation (Huang et al., 2025; Nasiriany et al., 2025; Wen et al., 2024; Gu et al., 2023; Fan et al., 2025a; Xu et al., 2024; Yu et al., 2024; Ma et al., 2024), STC flattens spatiotemporal signals and injects them into the diffusion transformer as conditioning features. It shares the backbone and training hyperparameters of Inpainting Policy to assess the necessity of the Inpainting Policy design.

4.1 STUDY ON ACTION INPAINTING

As shown in Figure 4, we investigate two aspects of action reconstruction in the temporal Cartesian space: **(1) Information quantity.** We examine how the amount of visible control information affects action reconstruction by comparing structured constraints, represented by 3D points, with random

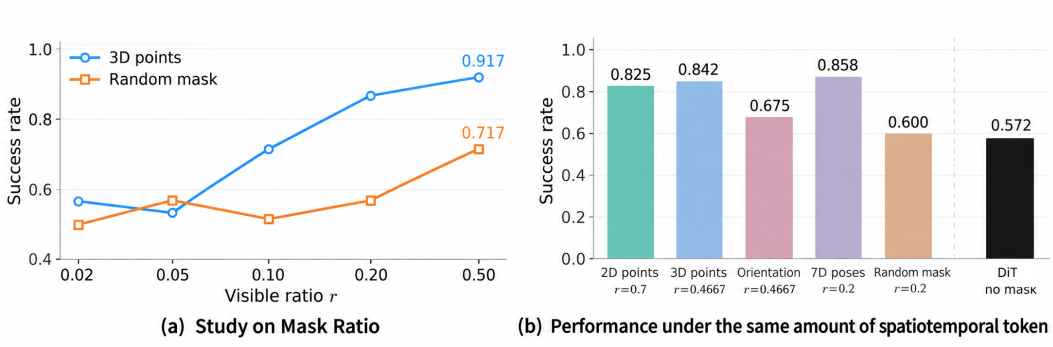


Figure 4: Effect of spatiotemporal information quantity and structure. (a) We vary the visible ratio to examine how the number of references affects task success. (b) We compare mask types with approximately equal numbers of spatiotemporal tokens, alongside a base policy (a DiT policy), to assess how constraint structure influences the performance of Inpainting Policy. Here, r denotes the visible ratio, i.e., the proportion of action information that is not masked in MAP.

constraints. **(2) Constraint category.** We examine the effect of mask structure while approximately matching the number of spatiotemporal tokens across different reference types. We conduct this study on ManiSkill using 100 demonstrations per task and a DiT backbone with approximately 49M parameters. We vary the visible ratio and mask type and evaluate each configuration on 80 held-out episodes per task. Detailed training settings and mask budgets are provided in Supplementary Sections 2.2 and 2.4, respectively.

Mask Ratio and Mask-Type Analysis. Figure 4(a) shows that increasing the visible-action ratio generally improves task success, indicating that additional known action components provide stronger guidance for reconstructing executable behaviors. For the structured 3D-point mask, success increases from 57.0% at a ratio of 0.02 to 91.7% at 0.50. Although the random mask also improves with more visible actions, its performance is less stable and reaches only 71.7% at the highest ratio. This gap suggests that structured control signals convey more task-relevant information than randomly selected action components.

Figure 4(b) compares different mask types under approximately the same number of visible spatiotemporal tokens. Across all mask types, IP outperforms the unconditioned DiT policy. Positional and full-pose constraints provide the strongest guidance, whereas orientation-only constraints are less informative for task completion. The random-mask result further shows that semantically structured action references are more effective than an equivalent number of arbitrary visible tokens.

4.2 Unified Control via Joint Training

In this section, as shown in Table 1, we investigate whether a single policy can follow diverse control references through joint training on versatile spatiotemporal masks. We assess task completion and reference adherence through success rate and control error. We evaluate joint training of the Inpainting Policy across diverse constraint types in two settings: single-task policies with a DiT backbone on ManiSkill, and a shared language-conditioned multitask policy with a U-Net backbone on LIBERO-10. IP-RePaint is applied directly at inference time to the baseline DP and DiT policies for Versatile control.

Joint-Training. On LIBERO-10, we adopt **two-stage joint training**. First, we train a shared policy across all ten tasks with random masks; second, we continue training with a mixture of structured masks with different reference modalities and sparsity levels. At evaluation time, Mix0 serves as an unseen mask type. We apply **mask-wise data augmentation** by pairing each demonstration with multiple mask configurations, yielding diverse reference signals with a limited number of demonstrations. See Supplementary 2.2 and 2.4 for the training settings and Mix0 construction, and 3.4 for the ablation study.

Table 1: **Unified control via joint training.** Each policy checkpoint is evaluated across reference types using the average top-three picks of success rate and control error (CE). ManiSkill uses a separate policy for each task, whereas LIBERO-10 uses a shared multitask policy.

	Multi-task (LIBERO-10)		Single-task (ManiSkill)				
Method	**Avg. SR (%) ↑**	**CE(all)↓**	**Lift Peg SR (%)↑**	**Pick Cube SR (%)↑**	**Poke Cube SR (%)↑**	**Avg. SR (%) ↑**	**CE(all)↓**
DP	60.00	—	86.33	69.33	69.67	75.11	—
DiT Policy	59.67	—	94.67	87.33	**93.00**	91.67	—
STC	50.67	0.060	72.67	65.67	91.67	76.67	0.041
IP	**64.67**	0.045	**100.00**	**90.33**	87.67	**92.67**	0.023
IP-RePaint	51.33	**0.026**	95.80	82.13	62.20	80.04	**0.014**

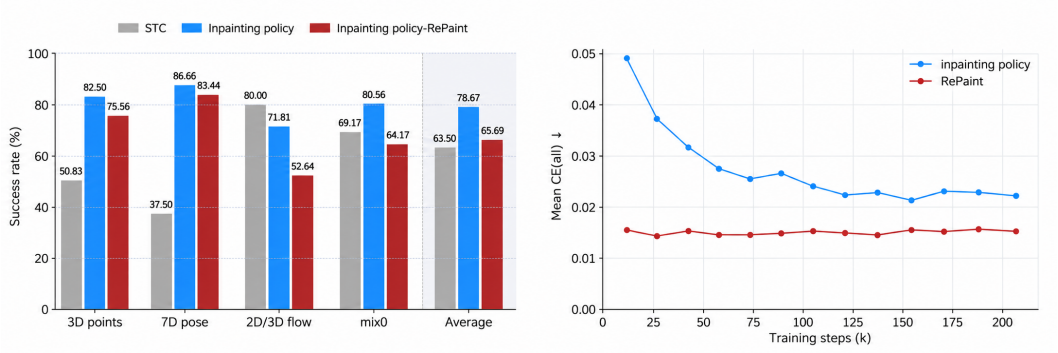


Figure 5: **Mask-type breakdown in joint training.** Left: We compare success rates across reference types, including the unseen mixed mask `Mix0`, to evaluate controllability and compositional generalization. Right: Control error decreases as training progresses.

Joint-Training Performance. Table 1 shows that STC performs worse than its corresponding base policies, indicating that directly conditioning action generation on control signals can interfere with task SR. In contrast, Inpainting Policy improves task completion while achieving lower control error, performing well in both multi-task and single-task settings and achieving the highest average success rates of 64.67% and 92.67%, respectively. The training-free RePaint variant further reduces CE with relative reductions of 42.2% and 39.1%, respectively, demonstrating a trade-off between control accuracy and task completion.

Mask-Type Breakdown and Training Dynamics. Figure 5 shows that Inpainting Policy provides the strongest average control performance across all control types, especially on the unseen `Mix0` mask (80.56%). STC performs strongly on dense 2D/3D flow (80.00%), but degrades sharply under sparse constraints. The right plot shows that Inpainting Policy progressively improves its control capability during training, with CE decreasing from approximately 0.049 to 0.022. In contrast, RePaint maintains the lowest CE, around 0.014–0.015, across all checkpoints. Notably, since the `Mix0` composition is absent from the training set, its high success rate indicates that the policy can **compose learned mask patterns** without additional finetuning.

4.3 Control with Real-World References

Real World Setting. For real-world experiments, we use Franka R3 robotic arms with two-finger grippers. One calibrated RealSense D455 camera provides the third-person view, and one ZED-M camera provides the wrist view. The robotic arm operates at a policy frequency of 15 Hz using absolute end-effector commands. We train task-specific Inpainting Policies (IP) and Diffusion Policy (DP) as the base policy. See Supplementary Section 2.6 for real-world implementation details.

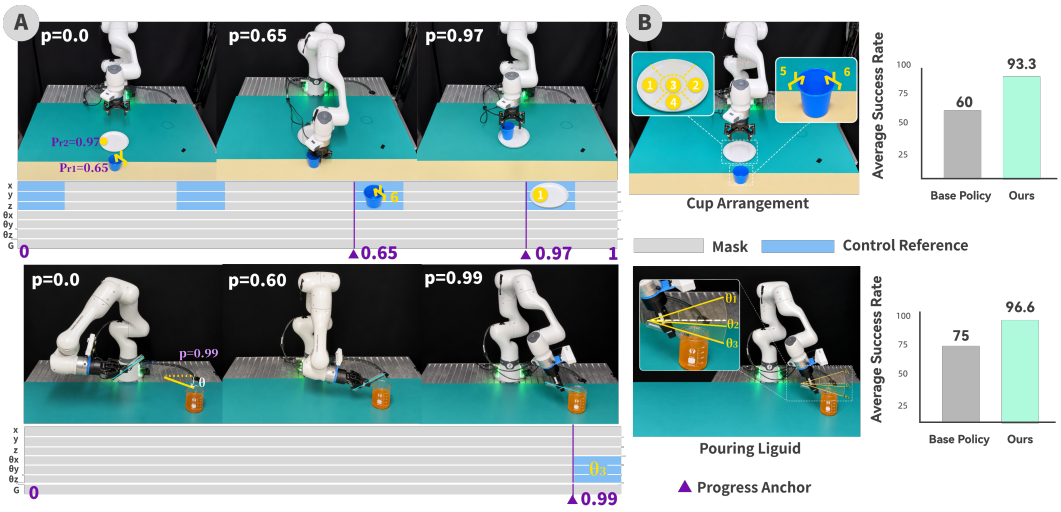


Figure 6: **Real-world spatiotemporal editing.** Here, p_r denotes the progress associated with a user-provided reference, whereas p denotes the robot's current execution progress. The rollouts and corresponding MAP masks illustrate how these references guide execution.

Human-in-the-Loop Control. The **Robotic Spatiotemporal Editing** system accepts a variable number of user-provided references, each associated with a reference type and a progress value p' that specifies when the constraint should be applied. The system converts all references into a unified **Masked Action Space (MAP)** for policy inference.

Tasks. We evaluate whether IP can complete the tasks while successfully satisfying additional spatial and orientational objectives with references including keypoints, flows, poses, and orientations, under both sparse and dense configurations. In the cup-arrangement task, these references control how the cup is grasped and where it is placed on the plate. In the liquid-pouring task, orientation references specify the tube's tilt angle and thereby regulate the dispensed volume.

Result. As shown in Figure 6, with three or more reference points, the policy achieves success rates above 93%, while the cup-to-reference-point distance ranges from 18.2 to $63.7\,\mathrm{mm}$, with an average of $44.3\,\mathrm{mm}$. In the liquid-pouring task, the tube-tilt error remains below $8°$, with a mean of $7.8°$. These results show that providing explicit control constraints substantially improves task completion and reference adherence: denser references produce smoother motion and smaller deviations from the target. In contrast, when only a single reference is provided at the final progress point, the task completion rate falls below 30%, revealing a limitation of sparse late-stage supervision and an important direction for improving the subsequent editing system. Supplementary Section 2.6 provides additional recorded SR and normalized CE results (Tables 3–5), plus deployment tips.

5 Conclusion and Future Work

Conclusion. We presented a unified framework for spatiotemporal action editing that augments policies with DiT or U-Net backbones with a unified control interface over heterogeneous spatiotemporal references. Without requiring additional control-specific data or increasing the backbone training budget, our method preserves the original task completion ability while enabling users to edit and compose spatiotemporal action constraints through masked action inpainting. Our inpainting formulation is model-agnostic and may be compatible with a broad range of foundation action models, including WAM and flow-matching-based VLAs, which we leave for future empirical study.

Limitations. Our current inpainting implementation may not be optimal; alternatives such as CFG and ControlNet-style conditioning remain unexplored. Future work will study large-scale BERT-style random-mask pretraining for foundation action models and evaluate zero-shot editing under more unstructured control specifications.

AI USE DISCLOSURE

In this work, we used generative AI tools to assist with translating and polishing the manuscript, drafting and editing parts of the paper, improving the readability of the presentation, modifying scientific figures, interpreting experimental results, and writing and modifying code.

We did not use generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, write proofs, propose or refine hypotheses, design or provide feedback on the research methodology or experiments, clean or reformat datasets, or perform qualitative or thematic data analysis.

All AI-assisted content was reviewed and verified by the authors. The authors take full responsibility for the final content of this submission, including all text, claims, experimental results, code, and figures.

ETHICS STATEMENT

This work studies controllable robot manipulation using public simulation benchmarks and controlled real-world robotic experiments. It does not involve human subjects, personal data, sensitive information, or high-risk applications. The real-world experiments were conducted in a controlled laboratory environment following standard robot-safety procedures. We have read and followed the ICLR Code of Ethics and take responsibility for the ethical conduct of this research.

REPRODUCIBILITY STATEMENT

To support reproducibility, the supplementary material describes the masked-action formulation and inference procedure in *Additional Method Details*, and the policy architectures, training settings, mask construction, evaluation metrics, and real-world setup in *Additional Implementation Details*. The *Additional Experiments* section reports the settings and evaluation protocols for the encoder, window-length, progress-module, training-schedule, and progress-alignment comparisons. These descriptions distinguish comparisons with matched settings from historical comparisons with differing training or evaluation conditions, and identify results aggregated across reference configurations and checkpoints.

REFERENCES

Dhruv Agrawal, Dominik Borer, Luca Vögeli, Robert Sumner, Martin Guay, and Jakob Buhmann. Interactive generative motion editing via scheduled inpainting. *arXiv preprint arXiv:2607.29133*, 2026.

Hadi Alzayer, Wenlong Huang, Haonan Chen, Christopher Luey, Lvmin Zhang, Maneesh Agrawala, Gordon Wetzstein, Li Fei-Fei, Yilun Du, Jiajun Wu, et al. Masked visual actions for unified world modeling. *arXiv preprint arXiv:2607.19343*, 2026.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Kevin Black, Manuel Y Galliker, and Sergey Levine. Real-time execution of action chunking flow policies, 2025. *URL https://arxiv. org/abs/2506.07339*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2026. URL https://arxiv.org/abs/2410.24164.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023. URL https://arxiv.org/abs/2212.06817.

Lawrence Yunliang Chen, Kush Hari, Karthik Dharmarajan, Chenfeng Xu, Quan Vuong, and Ken Goldberg. Mirage: Cross-embodiment zero-shot policy transfer with cross-painting. *arXiv preprint arXiv:2402.19249*, 2024.

Qianzhong Chen, Justin Yu, Mac Schwager, Pieter Abbeel, Yide Shentu, and Philipp Wu. Sarm: Stage-aware reward modeling for long horizon robot manipulation, 2026a. URL https://arxiv.org/abs/2509.25358.

Ruoqu Chen, Feixiang Ruan, Liu Cao, Zihao Wang, Botian Xu, Shiqin Tong, Jiajun Liu, Mingzhi Pei, Chenyu Zhang, Wanli Xing, Kaifeng Zhang, and Mengdi Xu. Dex-x: Learning visual-tactile dexterous manipulation from human videos with simulated interaction, 2026b. URL https://arxiv.org/abs/2609.07747.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Shichao Fan, Quantao Yang, Yajie Liu, Kun Wu, Zhengping Che, Qingjie Liu, and Min Wan. Diffusion trajectory-guided policy for long-horizon robot manipulation. *IEEE Robotics and Automation Letters*, 2025a.

Yiguo Fan, Pengxiang Ding, Shuanghao Bai, Xinyang Tong, Yuyang Zhu, Hongchao Lu, Fengqi Dai, Wei Zhao, Yang Liu, Siteng Huang, Zhaoxin Fan, Badong Chen, and Donglin Wang. Long-vla: Unleashing long-horizon capability of vision language action model for robot manipulation, 2025b. URL https://arxiv.org/abs/2508.19958.

Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 39(5):3929–3945, 2023.

Xiaolin Fang, Bo-Ruei Huang, Jiayuan Mao, Jasmine Shone, Joshua B Tenenbaum, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Kalm: Keypoint abstraction using large models for object-relative imitation learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8307–8314. IEEE, 2025.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, Qianli Ma, Seungjun Nah, Loic Magne, Jiannan Xiang, Yuqi Xie, Ruijie Zheng, Dantong Niu, You Liang Tan, K. R. Zentner, George Kurian, Suneel Indupuru, Pooya Jannaty, Jinwei Gu, Jun Zhang, Jitendra Malik, Pieter Abbeel, Ming-Yu Liu, Yuke Zhu, Joel Jang, and Linxi "Jim" Fan. Dreamdojo: A generalist robot world model from large-scale human videos, 2026. URL https://arxiv.org/abs/2602.06949.

Jiayuan Gu, Sean Kirmani, Paul Wohlhart, Yao Lu, Montserrat Gonzalez Arenas, Kanishka Rao, Wenhao Yu, Chuyuan Fu, Keerthana Gopalakrishnan, Zhuo Xu, et al. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. *arXiv preprint arXiv:2311.01977*, 2023.

Ce Hao, Kelvin Lin, Zhiwei Xue, Siyuan Luo, and Harold Soh. Disco: Language-guided manipulation with diffusion policies and constrained inpainting. *IEEE Robotics and Automation Letters*, 2025.

Kefeng Huang, Tingguang Li, Yuzhen Liu, Zhe Zhang, Jiankun Wang, and Lei Han. Vlm-tdp: Vlm-guided trajectory-conditioned diffusion policy for robust long-horizon manipulation. *arXiv preprint arXiv:2507.04524*, 2025.

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, Loic Magne, Ajay Mandlekar, Avnish Narayan, You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Yinzhen Xu, Xiaohui Zeng, Kaiyuan Zheng, Ruijie Zheng, Ming-Yu Liu, Luke Zettlemoyer, Dieter Fox, Jan Kautz, Scott Reed, Yuke Zhu, and Linxi Fan. Dreamgen: Unlocking generalization in robot learning through video world models, 2025. URL https://arxiv.org/abs/2505.12705.

Yuheng Ji, Huajie Tan, Jiayu Shi, Xiaoshuai Hao, Yuan Zhang, Hengyuan Zhang, Pengwei Wang, Mengdi Zhao, Yao Mu, Pengju An, et al. Robobrain: A unified brain model for robotic manipulation from abstract to concrete. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1724–1734. IEEE, 2025.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model, 2025a. URL https://arxiv.org/abs/2503.00200.

Yanbang Li, Ziyang Gong, Haoyang Li, Xiaoqi Huang, Haolan Kang, Guangping Bai, and Xianzheng Ma. Robotic visual instruction, 2025b. URL https://arxiv.org/abs/2505.00693.

Fangchen Liu, Kuan Fang, Pieter Abbeel, and Sergey Levine. Moka: Open-world robotic manipulation through mark-based visual prompting. *arXiv preprint arXiv:2403.03174*, 2024.

Mingyu Liu, Zheng Huang, Xiaoyi Lin, Muzhi Zhu, Canyu Zhao, Yating Wang, Haoyi Zhu, Hao Chen, and Chunhua Shen. Gae: Unleashing physical potential of vlm with generalizable action expert. In *Forty-third International Conference on Machine Learning*, 2025.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models, 2022. URL https://arxiv.org/abs/2201.09865.

Xiao Ma, Sumit Patidar, Iain Haughton, and Stephen James. Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18081–18090. IEEE, 2024.

Adithyavairavan Murali, Balakumar Sundaralingam, Yu-Wei Chao, Wentao Yuan, Jun Yamada, Mark Carlson, Fabio Ramos, Stan Birchfield, Dieter Fox, and Clemens Eppner. Graspgen: A diffusion-based framework for 6-dof grasping with on-generator training, 2025. URL https://arxiv.org/abs/2507.13097.

Soroush Nasiriany, Sean Kirmani, Tianli Ding, Laura Smith, Yuke Zhu, Danny Driess, Dorsa Sadigh, and Ted Xiao. Rt-affordance: Affordances are versatile intermediate representations for robot manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8249–8257. IEEE, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. URL https://arxiv.org/abs/2212.09748.

Michael Piseno, Guy Tevet, and C Karen Liu. Cloak: Zero-shot cross-embodiment manipulation by masking the end-effector from the vla. *arXiv preprint arXiv:2606.22836*, 2026.

Davis Rempe, Mathis Petrovich, Ye Yuan, Haotian Zhang, Xue Bin Peng, Yifeng Jiang, Tingwu Wang, Umar Iqbal, David Minor, Michael de Ruyter, et al. Kimodo: Scaling controllable human motion generation. *arXiv preprint arXiv:2603.15546*, 2026.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

Pengcheng Wang, Kaiwen Hong, Chensheng Peng, Katherine Driggs-Campbell, Masayoshi Tomizuka, Chenfeng Xu, and Chen Tang. Discretertc: Discrete diffusion policies are natural asynchronous executors. *arXiv preprint arXiv:2604.25050*, 2026a.

Tingwu Wang, Olivier Dionne, Michael De Ruyter, David Minor, Davis Rempe, Kaifeng Zhao, Mathis Petrovich, Ye Yuan, Chenran Li, Zhengyi Luo, et al. Motionbricks: Scalable real-time motions with modular latent generative model and smart primitives. *ACM Transactions on Graphics (TOG)*, 45(4):1–22, 2026b.

Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning, 2024. URL https://arxiv.org/abs/2401.00025.

Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface, 2024. URL https://arxiv.org/abs/2407.15208.

Rongtao Xu, Jian Zhang, Minghao Guo, Youpeng Wen, Haoting Yang, Min Lin, Jianzheng Huang, Zhe Li, Kaidong Zhang, Liqiong Wang, Yuxuan Kuang, Meng Cao, Feng Zheng, and Xiaodan Liang. A0: An affordance-aware hierarchical model for general robotic manipulation, 2026. URL https://arxiv.org/abs/2504.12636.

Dongjie Yu, Hang Xu, Yizhou Chen, Yi Ren, and Jia Pan. Bikc: Keypose-conditioned consistency policy for bimanual robotic manipulation, 2024. URL https://arxiv.org/abs/2406.10093.

Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspiar Singh, Clayton Tan, Dee M, Jodilyn Peralta, Brian Ichter, Karol Hausman, and Fei Xia. Scaling robot learning with semantically imagined experience, 2023. URL https://arxiv.org/abs/2302.11550.