ChronoFlow-Policy
Unifying Past-Current-Future Interaction Flow in Visuomotor Policy Learning

Bokai Lin[1,2*], Yifu Xu[1*], Xinyu Zhan[1], Hongjie Fang[1], Jialin Tian[1], Fu-Cheng Zhang[2], Yong-Lu Li[1,2], Cewu Lu[1,2,3], and Lixin Yang[1✉]

[1] Shanghai Jiao Tong University, China
[2] Shanghai Innovation Institute, China
[3] Noematrix, China
19821172068@sjtu.edu.cn, siriusyang@sjtu.edu.cn

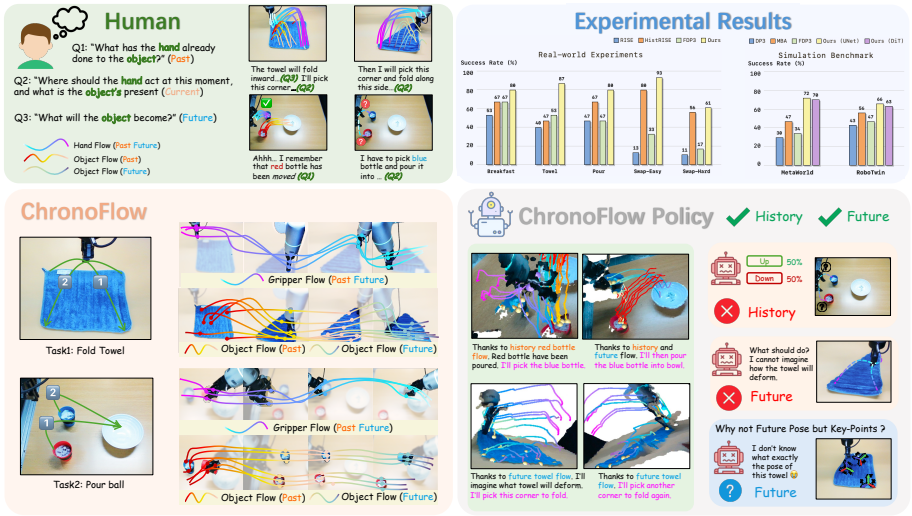


Fig. 1: Motivation and Overview of ChronoFlow. **Human** intuition solves manipulation tasks by reasoning over *past–current–future* interaction: recalling how the hand and object have moved, perceiving their current relation, and anticipating how the object should move next. Inspired by this temporal reasoning, **ChronoFlow** provides an intermediate representation for robot that encodes unified object-gripper keypoint flows across time. **ChronoFlow-Policy** leverages ChronoFlow as a co-training objective to learn actions, improving performance on both simulation and real-world manipulation.

Abstract. Visual signals play a crucial role in policy learning by enabling models to capture object motion and interaction dynamics. Just as humans reason about actions using both past experience and anticipated outcomes, effective policies should integrate past interactions with future predictions. However, existing visuomotor policies typically model either

*Equal contribution. ✉Corresponding author.

historical context or future dynamics in isolation, lacking a unified temporal representation of interaction dynamics. In this work, we introduce **ChronoFlow**, a temporally unified representation that captures **past, current, and future** interaction dynamics through sparse 3D keypoints of both objects and the gripper. Based on this representation, we propose **ChronoFlow-Policy**, a diffusion-based visuomotor policy that jointly learns ChronoFlow and action sequences through a co-training objective. Experiments on 14 simulated tasks and 5 real-world manipulation tasks demonstrate that ChronoFlow-Policy consistently outperforms strong diffusion-policy baselines and improves robustness in long-horizon and non-Markovian manipulation scenarios. Code and models will be released at `https://github.com/The-kamisato-Sii/ChronoFlow-Policy`.

Keywords: Visuomotor Policy · Spatial-Temporal Representation

1 Introduction

Recent advances in world action models have begun to couple action generation with predictive visual modeling. For example, UWM [42] and DreamZero [35] use structured visual prediction as auxiliary supervision, indicating that action generation can be improved by temporally aligned visual targets.

A complementary question is what form of visual supervision is better suited for embodied manipulation. Compared with image-space prediction alone, depth-aware 3D representations lift visual supervision into physical space. For example, object poses and scene point flows can describe how task-relevant entities move and interact over time. In visuomotor policy learning, existing efforts along this direction mainly follow two routes. One route predicts **future state evolution**, using object poses [9, 25] or scene-level point cloud dynamics [11, 16] to provide foresight for action generation. The other route incorporates **historical context**, motivated by the non-Markovian nature of many manipulation tasks [2, 3, 28], and uses past observations for long-horizon reasoning [3, 21, 41] . Despite their complementary roles, future prediction and history modeling are typically studied separately.

In this work, we propose **ChronoFlow**, a temporally unified flow representation that spans **past**, **current** and **future**. ChronoFlow represents **both objects and the gripper** (interaction-centric) as sparse 3D keypoints evolving over time, capturing their coupled motion in a shared spatial frame. ChronoFlow offers two practical advantages:

1. it is supported by recent advances in 3D point tracking [32,39] (*e.g.* D4RT [40]), which make it feasible to extract reliable historical traces from RGBD videos;
2. it focuses on task-relevant object-gripper keypoints, providing a compact representation that filters redundant scene motion while preserving interaction structure.

Based on ChronoFlow, we propose **ChronoFlow-Policy**, which couples action generation with ChronoFlow prediction in a unified diffusion framework.

By requiring the diffusion backbone to recover interaction-flow trajectories, the auxiliary objective encourages the shared latent representation to encode temporally grounded manipulation dynamics rather than action labels alone. The policy then decodes actions from this shared representation using a lightweight transformer-based decoder.

We evaluate ChronoFlow-Policy on 14 simulated tasks from MetaWorld [37] and RoboTwin 2.0 [4], together with 5 real-world manipulation tasks. Our method is compared against representative visuomotor policies, including strong 3D imitation policies like DP3 [38] and RISE [29]. We further compare with history-aware approaches such as HistRISE [3], as well as policies that incorporate future dynamics modeling, such as MBA [25] and 3D-FDP [16].

ChronoFlow-Policy achieves consistent performance improvements across both simulation and real-world tasks. In particular, in the real-world tasks, *Swap-Easy* and *Swap-Hard* require the policy to remember earlier object state when selecting later actions; historical ChronoFlow traces provide this memory and improve stage-dependent decision making. *Fold Towel* further shows that ChronoFlow can model deformable-object dynamics through interaction flows. On the long-horizon task *Prepare Breakfast*, ChronoFlow-Policy maintains strong performance across sequential stages, indicating that interaction-flow prediction helps preserve task progress during multi-step execution.

Our contribution are threefolds:

- We propose ChronoFlow, a compact interaction-centric 3D keypoint representation that unifies past-current-future dynamics.
- We propose ChronoFlow-Policy, a diffusion-based visuomotor policy that couples ChronoFlow prediction with action generation through a co-training objective, enabling actions to be decoded from a shared latent representation.
- We design a suite of long-horizon, non-Markovian, and non-rigid real-world manipulation tasks to assess temporal interaction modeling under stage-dependent decisions, deformable-object dynamics, and multi-step execution. ChronoFlow-Policy shows consistent gains across these settings, demonstrating the practical value of unified past-current-future interaction modeling.

2 Related work

Visuomotor Policy with Future Integration. Recent work in visuomotor policy learning increasingly incorporates future representations — which estimate how the scene may evolve — as auxiliary supervisions to improve action prediction. Some approaches predict future observations directly in the visual domain, where policies are conditioned on predicted frames or latent video representations. Such methods employ video diffusion models [10, 17] or jointly model latent video features with actions in a unified sequence [15, 35]. While effective at capturing long-horizon temporal patterns, these representations remain entangled with appearance information and often lack explicit spatial structure, which can introduce redundant signals unrelated to physical interaction.

Another line of work models future dynamics using more structured and abstract representations. Some methods predict object pose trajectories [25] or condition policies on sequences of object poses [9], enabling reasoning over object motion in SE(3). However, pose-based representations are less suitable for deformable objects and complex multi-object interactions. Other works represent future dynamics using object-level motion fields, such as object flow [1, 5, 11, 24, 31, 33]. Some approaches also model gripper flow to capture end-effector motion [8]. However, these methods typically model object and gripper motion separately, without explicitly capturing their interaction dynamics. More recently, several methods adopt scene flow to estimate dense point trajectories of the entire scene [11,16,30], using optical flow [12,23] or point tracking [32,39]. While expressive, such dense representations may include redundant motion signals unrelated to task-relevant interactions.

In contrast, we introduce an interaction-centric keypoint flow representation that jointly models past and future trajectories of both objects and grippers. By focusing on interaction-relevant keypoints, our formulation avoids redundant information while explicitly capturing gripper-object dynamics, enabling temporally consistent reasoning for complex manipulation.

Visuomotor Policy with History Integration. Prior work explores different ways to encode historical observations for visuomotor policies. Video-based representations capture long temporal horizons but incur high computational cost and redundancy [10, 14], while keyframe-based or compressed representations improve efficiency at the risk of missing subtle temporal transitions [7, 26, 36]. Hybrid designs combine multiple abstraction levels, yet often preserve visually salient but task-irrelevant information [13,22,27,34]. More recently, several methods like TraceVLA [41] and HistRISE [3] inject historical information through object-centric representations, for example by tracking object points over time and using them as historical context for policy learning. Most existing history representations focus on encoding past observations as contextual inputs for policy learning. However, these representations typically do not explicitly model object-gripper interactions, which play a central role in many manipulation tasks.

On the contrary, our method explicitly extracts interaction-centric keypoints that capture object-gripper contact and motion patterns. Moreover, beyond encoding history, we also predict future interaction trajectories and use them as a co-training objective. This design provides structured supervision on how past interactions evolve into future states, encouraging the model to learn unified past-current-future interaction dynamics for manipulation.

3 Methodology

3.1 Problem Setup

We study visuomotor policy learning for robot manipulation from demonstrations. At each timestep t, the robot receives an observation $\boldsymbol{o}_t = (\boldsymbol{p}_t, \boldsymbol{q}_t)$. Here

$\boldsymbol{p}_t \in \mathbb{R}^{N\times 6}$ denotes a scene-level point cloud captured from a single RGB-D camera, where each point is represented by its 3D coordinates and color (x, y, z, r, g, b). The number of points N may vary across timesteps. When available, $\boldsymbol{q}_t \in \mathbb{R}^{d_q}$ denotes the robot proprioceptive state, such as joint positions or end-effector pose. Given the current observation, the policy predicts a sequence of future low-level actions over a fixed horizon H, $\boldsymbol{o}_t \mapsto \boldsymbol{a}_{t:t+H}$. where each $\boldsymbol{a}_\tau \in \mathbb{R}^{d_a}$ corresponds to continuous control commands, such as end-effector pose increments.

Learning such a policy from demonstrations is challenging in complex manipulation scenarios. Observations from a single timestep provide only partial information about the interaction state, while many manipulation tasks exhibit strong temporal dependencies due to contact dynamics and staged object interactions. As a result, policies trained solely on instantaneous observations often struggle to capture the underlying spatiotemporal structure of object–robot interactions. In this work, we address this challenge by introducing an interaction-centric representation that explicitly models the temporal evolution of task-relevant motions.

3.2 ChronoFlow: Past-Current-Future Interaction Flow

To explicitly model interaction dynamics in manipulation tasks, we introduce **ChronoFlow**, a unified representation of past-current-future interaction based on sparse 3D keypoint trajectories.

At each timestep t, ChronoFlow represents the interaction state using a set of keypoints defined on both the robot gripper and task-relevant objects. Specifically, we denote the gripper keypoints as $\boldsymbol{P}_t^g \in \mathbb{R}^{N_g\times 3}$ and the object keypoints as $\boldsymbol{P}_t^o \in \mathbb{R}^{N_o\times 3}$, where N_g and N_o are the numbers of gripper and object keypoints, respectively. Their union $\boldsymbol{P}_t = \boldsymbol{P}_t^g \cup \boldsymbol{P}_t^o$ provides a sparse abstraction of the physical interaction state at time t.

Gripper Keypoints Flow. For the robot gripper, we predefine N_g canonical keypoints on the gripper geometry. These keypoints are rigidly attached to the tool center point (TCP). Given the TCP pose at each timestep, we compute the 3D positions of all gripper keypoints via rigid transformation, producing a temporally consistent gripper trajectory in the world coordinate frame.

Object Keypoints Flow. For manipulated objects, we first segment task-relevant objects from the initial observation using 3D segmentation SAM-2 [19]. For each segmented object, we apply Farthest Point Sampling (FPS) to obtain a dense set of candidate keypoints. Aggregating across objects yields a pool of object keypoints that sparsely represent the scene geometry. These keypoints are tracked over time using TAPIP3D [39], producing temporally consistent object-centric trajectories.

The resulting ChronoFlow trajectories provide a compact representation of interaction dynamics. Since the gripper flow is structurally aligned with robot actions, perturbing ChronoFlow implicitly injects noise into both the future object motion and action. This interaction-centric formulation naturally leads to

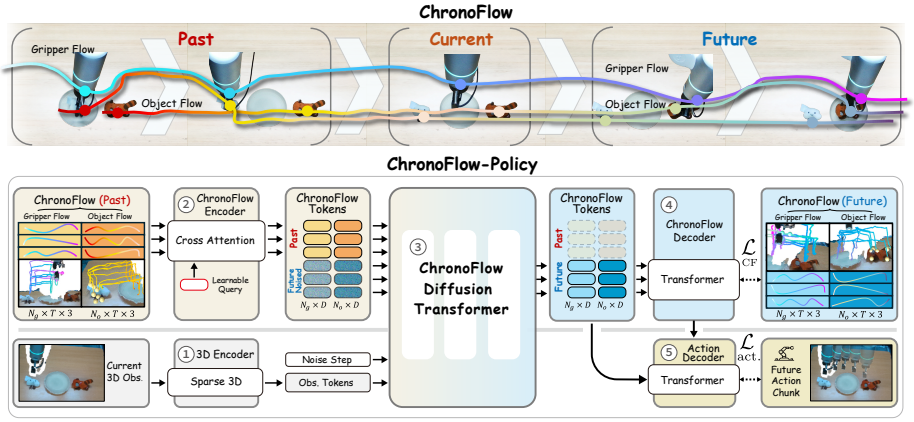


Fig. 2: Overview of **ChronoFlow-Policy**. **Top:** Illustration of ChronoFlow: unified past-current-future object-gripper keypoint flows. **Bottom:** The network architecture.

joint flow matching. In the next section, we describe how these trajectories are incorporated into a joint flow-matching visuomotor policy.

3.3 ChronoFlow-Policy

Based on the ChronoFlow representation, we propose **ChronoFlow-Policy**, a diffusion-based visuomotor policy that explicitly models interaction dynamics and leverages them for action generation.

Overview. Instead of directly predicting actions from current visual observations, ChronoFlow-Policy first models the future evolution of interaction-centric keypoints and then generates robot actions conditioned on the predicted interaction trajectory. Formally, we factorize the policy as

$$\pi(\boldsymbol{a}_{t:t+H} \mid \boldsymbol{o}_t, \boldsymbol{P}_{:t}) = \pi_\phi(\boldsymbol{a}_{t:t+H} \mid \boldsymbol{P}_{t:t+H})\, \pi_\theta(\boldsymbol{P}_{t:t+H} \mid \boldsymbol{o}_t, \boldsymbol{P}_{:t}) \tag{1}$$

where $\boldsymbol{P}_{t:t+H}$ denotes the ChronoFlow trajectory describing the future motion, and $\boldsymbol{P}_{:t} = \{\boldsymbol{P}_\tau\}_{\tau<t}$ denotes the historical ChronoFlow trajectories constructed from past observations. The first term π_θ in Eq.1 models the evolution of interaction dynamics, while the second term π_ϕ maps the predicted interaction trajectory to robot actions.

Diffusion-based Interaction Modeling. To model the distribution $\pi_\theta(\boldsymbol{P}_{t:t+H} \mid \boldsymbol{o}_t, \boldsymbol{P}_{:t})$, we employ a conditional diffusion process over ChronoFlow trajectories. Starting from Gaussian noise, the model iteratively refines a noisy trajectory through a sequence of denoising steps:

$$\boldsymbol{P}^{k-1}_{t:t+H} = \alpha_k \boldsymbol{P}^{k}_{t:t+H} - \gamma_k \epsilon_\theta(\boldsymbol{P}^{k}_{t:t+H},\, \boldsymbol{o}_t,\, \boldsymbol{P}_{:t},\, k) + \sigma_k \mathcal{N}(0, I), \tag{2}$$

where ϵ_θ denotes the denoising network, $k \in (0, 1)$ denotes the diffusion timestep, and $\{\alpha_k, \gamma_k, \sigma_k\}$ define the diffusion noise schedule.

① ***Observation Encoder.*** Observation encoder maps the current 3D point cloud observation to a condition feature $\boldsymbol{c}_t = E_{\text{obs}}(\boldsymbol{o}_t)$. We use a PointNet-style point-cloud encoder in simulation and a sparse 3D encoder in real-world experiments. When proprioception is available, an MLP embeds it and fuses it with the point-cloud feature.

② ***ChronoFlow Encoder.*** ChronoFlow encoder converts historical ChronoFlow and the noisy future ChronoFlow into **ChronoFlow tokens**. Given the future ChronoFlow trajectory $\boldsymbol{P}_{t:t+H} \in \mathbb{R}^{(N_g+N_o)\times H\times 3}$, we perturb it with noise at flow-matching step k, obtaining $\boldsymbol{P}^k_{t:t+H}$. The multi-head cross-attention module (MCA in Eq. (3)) uses learnable interaction queries $\boldsymbol{Q} \in \mathbb{R}^{(N_g+N_o)\times D}$ to jointly encode the historical ChronoFlow $\boldsymbol{P}_{:t}$ and the noisy future ChronoFlow $\boldsymbol{P}^k_{t:t+H}$ into ChronoFlow tokens $\boldsymbol{Z}^k_t \in \mathbb{R}^{(N_g+N_o)\times D}$:

$$\boldsymbol{Z}^k_t = \text{MCA}\left(\boldsymbol{Q}, \left[\boldsymbol{P}_{:t}; \boldsymbol{P}^k_{t:t+H}\right]\cdot \boldsymbol{W}_K, \left[\boldsymbol{P}_{:t}; \boldsymbol{P}^k_{t:t+H}\right]\cdot \boldsymbol{W}_V\right), \tag{3}$$

where $\boldsymbol{W}_K$ and $\boldsymbol{W}_V$ are the key and value projection matrices, respectively. Each compact ChronoFlow token corresponds to one gripper or object keypoint. $\boldsymbol{Z}^k_t$ serves as the compact latent representation used by the flow-matching model $\epsilon_\theta\left(\boldsymbol{P}^k_{t:t+H}, \boldsymbol{o}_t, \boldsymbol{P}_{:t}, k\right)$ in Eq. 2.

③ ***Joint Diffusion Denoising.*** The backbone of ChronoFlow-Policy is a diffusion module, implemented either as a Unet or a Diffusion Transformer (DiT). Given the observation tokens, the noise timestep, and the ChronoFlow tokens $\boldsymbol{Z}^k_t$, the model performs conditional denoising in the ChronoFlow token space. This process progressively refines $\boldsymbol{Z}^k_t$ into clean interaction tokens $\boldsymbol{Z}^0_t$, which serve as the shared latent representation for subsequent ChronoFlow prediction and action decoding.

④ ***ChronoFlow Decoder.*** After denoising step, the ChronoFlow decoder converts the denoised tokens $\boldsymbol{Z}^0_t$ into future ChronoFlow using a lightweight Transformer decoder:

$$\hat{\boldsymbol{P}}_{t:t+H} = D_{\text{CF}}(\boldsymbol{Z}^0_t) \in \mathbb{R}^{(N_g+N_o)\times(H\cdot 3)}. \tag{4}$$

The output is reshaped into $\hat{\boldsymbol{P}}_{t:t+H} \in \mathbb{R}^{(N_g+N_o)\times H\times 3}$, representing future 3D trajectories of gripper and object keypoints.

⑤ ***Action Decoder.*** Action decoder predicts robot actions from the decoded ChronoFlow trajectory and the diffusion hidden state. We concatenate $\hat{\boldsymbol{P}}_{t:t+H}$ with the clean ChronoFlow tokens $\boldsymbol{Z}^0_t$ and feed them into a lightweight transformer decoder $\mathcal{A}_\phi(\cdot)$ for action prediction: $\hat{\boldsymbol{a}}_{t:t+H} = \mathcal{A}_\phi([\boldsymbol{Z}^0_t; \hat{\boldsymbol{P}}_{t:t+H}])$.

Training Objectives. The training objective follows the factorization in Eq. 1: we first supervise the diffusion backbone to recover ChronoFlow trajectories, and then align the learned interaction representation with action prediction.

To explicitly enforce interaction-centric dynamics learning, the denoising network ϵ_θ is trained using the standard diffusion objective:

$$\mathcal{L}_{\text{CF}} = \mathbb{E}_k \left[\|\epsilon - \epsilon_\theta(\boldsymbol{P}^k_{t:t+H}, \boldsymbol{o}_t, \boldsymbol{P}_{:t}, k)\|^2 \right]. \tag{5}$$

This ChronoFlow supervision encourages the backbone to model past-current-future consistent object-gripper interaction trajectories.

For action learning, directly using fully denoised trajectories would require completing the entire diffusion process for every optimization step. Considering that partially denoised hidden states already encode rich information [10, 17], we condition the action decoder on *partially denoised* ChronoFlow tokens $\boldsymbol{Z}_t^{k-1}$ during training. Additionally, to provide an explicit interaction cue, we reconstruct an *approximate* ChronoFlow trajectory $\tilde{\boldsymbol{P}}_{t:t+H}$ from an intermediate noisy diffusion state:

$$\tilde{\boldsymbol{P}}_{t:t+H} = \frac{1}{\sqrt{\bar{\alpha}_k}} \Big(\boldsymbol{P}^k_{t:t+H} - \sqrt{1-\bar{\alpha}_k}\, \epsilon_\theta(\boldsymbol{P}^k_{t:t+H},\, \boldsymbol{o}_t,\, \boldsymbol{P}_{:t},\, k) \Big). \tag{6}$$

The action decoder takes the partially denoised tokens $\boldsymbol{Z}_t^{k-1}$ and the approximate trajectory $\tilde{\boldsymbol{P}}_{t:t+H}$ as input, and is trained with an MSE loss against ground-truth actions:

$$\mathcal{L}_{\text{action}} = \mathbb{E}_k \left[\|\mathcal{A}_\phi([\boldsymbol{Z}_t^{k-1}; \tilde{\boldsymbol{P}}_{t:t+H}]) - \boldsymbol{a}_{t:t+H}\|^2 \right]. \tag{7}$$

The final objective combines action supervision with ChronoFlow supervision:

$$\mathcal{L} = \mathcal{L}_{\text{action}} + \lambda_{\text{CF}} \mathcal{L}_{\text{CF}}. \tag{8}$$

In this formulation, the ChronoFlow loss shapes the diffusion backbone into a unified spatiotemporal interaction model, while the action loss aligns the learned representation with task-level control objectives.

Inference. During inference, historical ChronoFlow trajectories $\boldsymbol{P}_{:t}$ are obtained asynchronously using a 3D tracker [39]. Starting from Gaussian noise $\boldsymbol{P}^K_{t:t+H}$, the ChronoFlow encoder and diffusion backbone iteratively produce clean interaction tokens $\boldsymbol{Z}_t^0$. The ChronoFlow and action decoders then generate the future trajectory $\hat{\boldsymbol{P}}_{t:t+H}$ and actions $\hat{\boldsymbol{a}}_{t:t+H}$.

4 Experiments

Experiment Overview. We evaluate ChronoFlow-Policy in both simulation and real-world manipulation. Simulation benchmarks provide controlled comparisons against action-only, scene-flow, and object-pose supervision, highlighting the effect of interaction-centric future modeling. Real-world experiments further test the full past-current-future formulation in long-horizon, deformable, and non-Markovian tasks.

Table 1: Comparison of methods by supervision, history, and future modeling.

Method	**Supervision**	**History**	**Future**	**Gain** (Sim. / Real)
DP3 / RISE	Action-only	×	×	– / –
HistDP3 / HistRISE	Action-only	✓	×	+2.2 / +25.8
3D-FDP	Dense scene flow	×	✓	+4.0 / +6.9
MBA	Object pose traj.	×	✓	+15.0 / +20.0†
CFP *w/o past*	ChronoFlow	×	✓	**+32.5** / +11.8
CFP	ChronoFlow	✓	✓	**+32.5 / +35.3**

† MBA real-world gain is computed only on *Prepare Breakfast*.

Tab. 1 summarizes the compared policy paradigms by supervision signal, history usage, future modeling, and overall gains. Action-only policies such as DP3 [38] and RISE [29] rely solely on action supervision; history-aware variants: HistRISE [3] add past observations without future prediction; 3D-FDP [16] and MBA [25] introduce future supervision through dense scene flow or object pose trajectories. In contrast, ChronoFlow-Policy (**CFP**) uses interaction-centric object–gripper ChronoFlow as the future prediction target, and the full model further incorporates historical ChronoFlow tokens. The **Gain** column reports absolute improvement over the corresponding base policy: DP3 in simulation and RISE in real-world experiments.

4.1 Simulation Experiments

Benchmarks.

- **MetaWorld** [37]. MetaWorld is a MuJoCo-based manipulation benchmark with a single gripper interacting with rigid and articulated objects, providing a controlled setting that is well suited for modeling gripper-object interactions and extracting interaction-centric keypoints.
- **RoboTwin** [4]. RoboTwin 2.0 is a dual-arm simulation framework for bimanual manipulation, offering diverse objects, tasks, and domain randomization, and enabling the study of coordinated interactions between two arms and objects in complex scenes.

Baselines. For simulation evaluations on MetaWorld and RoboTwin 2.0, we compare ChronoFlow-Policy with representative 3D visuomotor policies: 3D Diffusion Policy (**DP3**) [38], 3D Flow Diffusion Policy (**3D-FDP**) [16], and Motion Before Action (**MBA**) [25]. Since the simulation benchmarks are largely Markovian, CFP *w/o past* isolates the effect of future ChronoFlow supervision under the same Unet backbone. This comparison highlights the benefit of ChronoFlow over dense scene flow or object pose supervision. We implement CFP with both Unet1D [20] and DiT [18] diffusion backbones, denoted as **CFP (Unet)** and **CFP (DiT)** in the following tables.

Table 2: Success rates (%) on **MetaWorld**. Each column corresponds to one task. Results are reported as mean ± standard deviation over evaluation rollouts. Task abbreviations denote **shelf** (*Shelf Place*), **bin** (*Bin Picking*), **box** (*Box Close*), **peg** (*Peg Insert Side*), **hand** (*Hand Insert*), **soccer** (*Soccer*), and **sweep** (*Sweep Into*). **Bold**/blue shading indicates the best result and underline/light-blue shading indicates the second-best result.

Method	shelf	bin	box	peg	hand	soccer	sweep	**Avg.**
DP3 [38]	17 ± 10	34 ± 30	42 ± 3	69 ± 7	14 ± 4	18 ± 3	15 ± 5	30
3D-FDP [16]	16 ± 3	35 ± 9	56 ± 7	70 ± 8	17 ± 2	19 ± 4	25 ± 7	34
MBA [25]	73 ± 1	54 ± 23	56 ± 2	75 ± 5	10 ± 1	27 ± 3	34 ± 25	47
CFP (Unet)	63 ± 5	$\mathbf{94} \pm 1$	71 ± 1	$\mathbf{89} \pm 3$	$\mathbf{67} \pm 1$	$\mathbf{59} \pm 3$	60 ± 5	**72**
CFP (DiT)	$\mathbf{82} \pm 7$	70 ± 13	$\mathbf{75} \pm 5$	69 ± 5	60 ± 25	57 ± 5	$\mathbf{75} \pm 5$	70

Table 3: Success rates (%) on **RoboTwin 2.0**. Each column corresponds to one task. Task abbreviations denote **beat** (*Beat Block Hammer*), **mic** (*Handover Microphone*), **bell** (*Click Bell*), **card** (*Move Playingcard Away*), **cup** (*Place Empty Cup*), **clock** (*Click Alarmclock*), **bottles** (*Place Dual Bottles*). **Bold**/blue shading indicates the best result and underline/light-blue shading indicates the second-best result.

Method	beat	mic	bell	card	cup	clock	bottles	**Avg.**
DP3 [38]	39	64	66	32	27	54	18	43
3D-FDP [16]	58	65	80	**41**	27	44	16	47
MBA [25]	61	71	84	37	39	71	27	56
CFP (Unet)	**77**	**88**	81	39	**59**	68	**51**	**66**
CFP (DiT)	61	82	**93**	**41**	42	**79**	28	63

Protocols. During training, we adopt a unified temporal setting across all experiments: the action prediction horizon is fixed to 8 steps, the observation horizon is 1 step, and during inference only the first 4 predicted action steps are executed in a receding-horizon manner. Adhering to the protocol established in [16], each MetaWorld experiment is conducted across three trials with seed values 0, 1, and 2. The policy is evaluated over 20 episodes every 200 training epochs, and the mean of the top 5 success rates is reported. The final performance is computed as the mean and standard deviation across the three seeds. For RoboTwin 2.0, we evaluate the policy over 100 episodes every 500 training epochs and report the mean of the top 3 success rates. The large evaluation sample size yields stable results without requiring multiple random seeds.

Results. On MetaWorld, our CFP achieves clear improvements over prior methods across all tasks. As shown in Tab. 2, CFP (Unet) attains an average success rate of 72%, slightly outperforming CFP (DiT) (70%), substantially outperforming DP3 (30%), 3D-FDP (34%), and MBA (47%). The performance gain is particularly pronounced in interaction-intensive tasks such as *Hand-Insert* and *Sweep Into*. For example, on *Hand-Insert*, CFP (Unet) achieves 67% compared

to 14% (DP3) and 17% (3D-FDP), while on *Sweep Into* it reaches 60% versus 15% (DP3). We attribute these gains to the interaction-centric modeling of ChronoFlow. Methods such as 3D-FDP operate on full scene point clouds, which may obscure subtle gripper-object contact patterns within global geometry. In contrast, MBA relies solely on object pose supervision and lacks explicit gripper modeling, limiting its ability to reason about fine-grained contact transitions, especially in tasks requiring precise insertion or constrained manipulation like *Sweep Into*. ChronoFlow's joint modeling of gripper and object keypoints provides a more expressive and localized representation of interaction dynamics.

On RoboTwin 2.0 Tab. 3, our CFP also demonstrates consistent improvements in dual-arm manipulation. CFP (Unet) achieves an average success rate of 66%, surpassing DP3 (43%), 3D-FDP (47%), and MBA (56%), slightly overcomes CFP (DiT) (63%). Despite using only a single observation frame, CFP performs strongly in tasks requiring accurate spatial localization, such as *Beat Block Hammer* and *Click Alarmclock*, where interaction-based modeling enables more precise alignment and timing. Furthermore, in long-horizon dual-arm tasks such as *Handover Microphone*, CFP effectively captures the object transition process between two grippers. The explicit modeling of gripper-object interaction flow allows the policy to better represent intermediate transfer states, which are difficult to encode using object-only or scene-level representations.

4.2 Real-World Experiments

In real-world experiments, we aim to answer the following research questions:

- **Q1:** Does history ChronoFlow improve non-Markovian performance?
- **Q2:** Does ChronoFlow policy gain benefits from future modeling?
- **Q3:** Can ChronoFlow Policy handle deformable object manipulation?
- **Q4:** Which design components of ChronoFlow are critical for real-world robustness (ablation)?

Platform. For real-world experiments, we use a Flexiv Rizon robotic arm equipped with a Robotiq 2F-85 gripper. Perception is provided by a single fixed top-down RGB-D camera (Intel RealSense D415), which captures the workspace point cloud for 3D perception. All real-world experiments are conducted with a 3D policy only, and no additional cameras or 2D visual inputs are used. For deployment efficiency, we run TAPIP3D [39] asynchronously with the policy. Tab. 4 shows that synchronous tracking reduces the average inference frequency to 0.93 Hz, whereas the asynchronous pipeline reaches 5.82 Hz. This design allows CFP to use historical ChronoFlow while avoiding 3D tracking as the runtime bottleneck.

Table 4: Inference frequency.

Method	Avg. Freq.
CFP w/o history track	12.18 Hz
CFP w/ TAPIP3D (Sync.)	0.93 Hz
CFP w/ TAPIP3D (Async.)	5.82 Hz

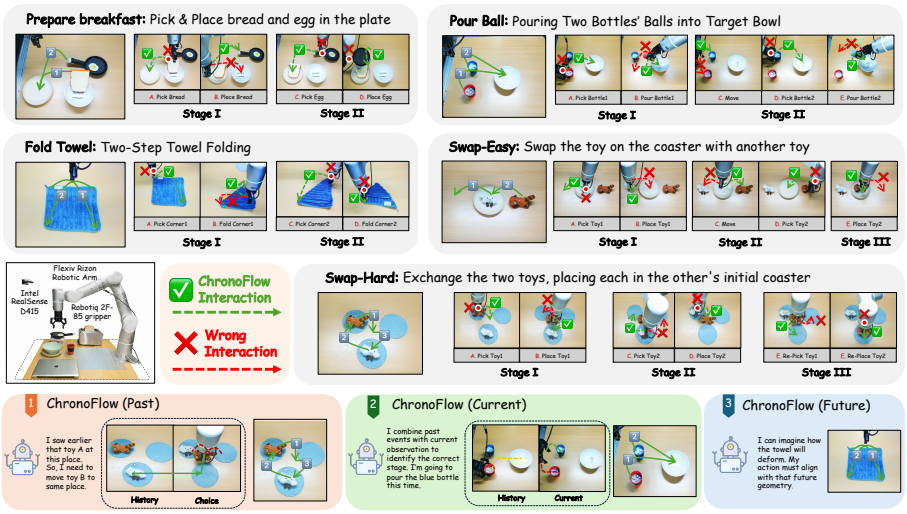


Fig. 3: Real-world manipulation tasks used for evaluation. We consider five real-world tasks, including *Prepare Breakfast*, *Pour Ball*, *Fold Towel*, *Swap-Easy*, and *Swap-Hard*. Each task is decomposed into multiple stages.

Tasks & Metrics. We select five real-world manipulation tasks for evaluation: *Prepare Breakfast* (long-horizon manipulation), *Fold Towel* (soft-body manipulation), and three non-Markovian tasks including *Swap Easy*, *Swap Hard*, and *Pour Ball*. To enable fine-grained analysis of long-horizon behavior, each task is decomposed into a sequence of semantically meaningful stages. Fig. 3 illustrates the task setups together with their corresponding stage definitions. These staged tasks evaluate the model's ability to remember **past** interactions, recognize the **current** manipulation stage, and plan **future** actions. Specifically, *Prepare Breakfast* consists of two stages: moving the bread and pouring the egg. *Pour Ball* contains two stages corresponding to successfully pouring balls into Bottle1 and Bottle2. *Fold Towel* contains two stages representing the first and second folding actions. *Swap-Easy* contains three stages: placing Toy1, picking up Toy2, and placing Toy2. *Swap-Hard* also contains three stages: placing Toy1 at the intermediate position, swapping Toy2, and finally picking up the intermediate Toy1 to place it at Toy2's original position.

We evaluate all real-world tasks using the one-attempt success rate, defined as whether the task is completed successfully within a single execution without resets. In addition, we record the completion rate of each stage to provide a more detailed analysis of task progress.

Baselines. We compare ChronoFlow-Policy with representative 3D point-cloud-based baselines: **RISE** [29], History-Aware RISE (**HistRISE**) [3], 3D Flow Diffusion Policy (**3D-FDP**) [16], and Motion Before Action (**MBA**) [25]. To study the role of historical information **(Q1)**, we additionally include an ablation

Table 5: Real-world success rates (%) across five manipulation tasks: **Breakfast** (*Prepare Breakfast*), **Towel** (*Fold Towel*), **Pour** (*Pour Ball*), **Swap-Easy**, and **Swap-Hard**. Each task is decomposed into sequential stages (I-III); detailed stage definitions are provided in Fig. 3. **Bold**/blue shading indicates the best performance and underline/light-blue shading indicates the second-best performance for each stage.

Task	**Breakfast**		**Towel**		**Pour**		**Swap-Easy**			**Swap-Hard**		
Stage	I	II	I	II	I	II	I	II	III	I	II	III
RISE [29]	73	53	80	40	80	47	87	27	13	89	17	11
3D FDP [16]	73	67	**87**	53	73	47	**93**	40	33	83	33	17
HistRISE [3]	87	67	**87**	47	80	67	**93**	80	80	**94**	89	56
CFP *w/o past*	**93**	67	**87**	**87**	**93**	60	**93**	33	20	**94**	22	11
CFP *(ours)*	**93**	**80**	**87**	**87**	87	**80**	**93**	**93**	**93**	**94**	**94**	**61**

variant, ChronoFlow-Policy without history (**CFP *w/o past***), which removes historical keypoints from ChronoFlow while keeping all other components unchanged. Since MBA relies on object pose trajectories, which are less suitable for history-dependent tasks and deformable-object manipulation, we report MBA on *Prepare Breakfast*, where object-level future motion can be defined more reliably. For real-world experiments, ChronoFlow-Policy is implemented with a Unet1D diffusion [17] backbone. For clarity, we denote this variant as **CFP** in the following tables. All methods operate on the same 3D point-cloud observations and are trained and evaluated under identical settings whenever applicable. Gains in Tab. 1 are computed relative to RISE, which serves as the action-only base policy with a strong 3D encoder. CFP *w/o past* isolates the benefit of future ChronoFlow supervision, while full CFP further evaluates whether historical object–gripper flows improve non-Markovian decisions.

Protocols. For each real-world task, we collect 50 expert demonstrations using the haptic device teleoperation [6] for training. All policies are trained on a workstation equipped with an NVIDIA RTX 3090. During evaluation, the workspace configuration is kept identical across all methods, and test initializations are uniformly sampled to ensure consistent coverage. Each policy is evaluated over 15 trials per task; for the more challenging *Swap-Hard* task, we increase the evaluation to 18 trials.

Results. Fig. 3 summarizes the real-world manipulation results, while Fig. 4a visualizes the interaction-centric keypoint flows predicted by ChronoFlow Policy during execution. We analyze the performance from three aspects: resolving non-Markovian dependencies (Q1), improving long-horizon task execution (Q2), and handling deformable object manipulation (Q3).

Modeling historical interaction improves non-Markovian manipulation (Q1). Tasks such as *Swap-Easy* and *Swap-Hard* require decisions that de-

pend on earlier interaction states rather than solely on the current observation. Methods without explicit history modeling often perform well at the beginning but fail to maintain task progress in later stages. For example, in *Swap-Easy*, RISE achieves 87% success in Stage I but drops sharply to 27% and 13% in Stages II and III. A similar pattern appears for 3D-FDP (93% → 40% → 33%) and CFP *w/o past* (93% → 33% → 20%). The same phenomenon is observed in *Swap-Hard*. In contrast, CFP maintains strong performance across all stages (93%, 93%, 93% in *Swap-Easy* and 94%, 94%, 61% in *Swap-Hard*), showing that historical ChronoFlow effectively resolves non-Markovian dependencies.

Future interaction modeling improves long-horizon tasks (Q2). On traditional long-horizon tasks such as *Prepare Breakfast*, CFP achieves consistently strong results across both stages. As shown in Tab. 5 and Tab. 6, our method CFP reaches 93% success in Stage I and improves to 80% in Stage II, outperforming RISE (73%, 53%), 3D-FDP (73%, 67%) and MBA (87%, 73%). These results show that joint past-current-future interaction modeling improves temporal reasoning, multi-step planning, and task progression.

Table 6:
Comparison with MBA.

Method	Breakfast	
	Stage I	**Stage II**
MBA	87%	73%
CFP	**93%**	**80%**

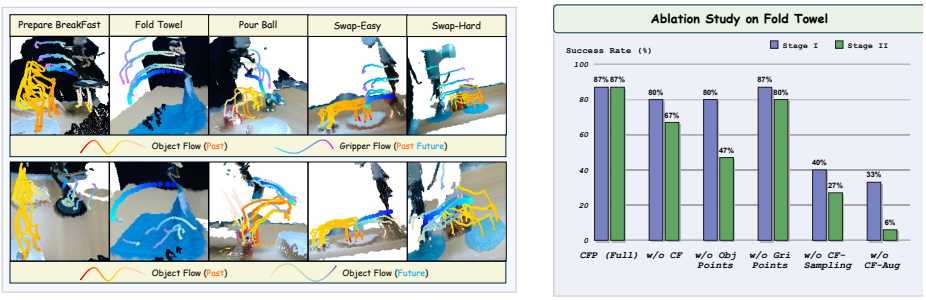
ChronoFlow Policy effectively handles deformable object manipulation (Q3). The *Fold Towel* task highlights the need to model deformable object dynamics. After the first fold, the towel exhibits highly variable geometry, making Stage II difficult for methods with limited object representations. RISE drops from 80% to 40%, often failing to localize or grasp the folded towel. MBA relies on object pose trajectories, but pose estimation for deformable objects is inherently ambiguous and can introduce substantial noise. In contrast, CFP models interaction-centric keypoint flows, which better preserve towel geometry under deformation. As a result, CFP achieves 87% success in both stages and substantially outperforms prior baselines in Stage II.

Ablations. ***Ablation studies confirm the importance of ChronoFlow design components (Q4).*** To assess the key design components of CFP, we conduct five ablation experiments on *Fold Towel*, as shown in Fig. 4b.

First, we separately ablate ChronoFlow supervision and flow inputs. Without ChronoFlow supervision (*w/o ChronoFlow*), Stage II performance drops from 87% to 67%. Removing object flows (*w/o ObjPoints*) causes a larger Stage II drop (87% → 47%) than removing gripper flows (*w/o GriPoints*, 87% → 80%), indicating that object flows are the primary signal for deformable-object tracking while gripper flows provide complementary cues.

Second, disabling keypoint sampling (*w/o Sampling*) degrades both stages (Stage I: 87% → 40%; Stage II: 87% → 27%), showing that random keypoint sampling improves spatial coverage and reduces overfitting to fixed tracks.

To improve robustness to viewpoint changes and sensing noise in real-world deployment, our full model applies the same spatial transformation jitter to both



(a) (b)

Fig. 4: Left: Predicted ChronoFlow in real-world tasks: gripper flows (top), object flows (bottom), and historical object flows (yellow/orange). **Right:** Ablation on *Fold Towel*. **CFP (Full)** denotes the full model. **w/o CF** removes ChronoFlow trajectory supervision. **w/o ObjPoints** and **w/o GriPoints** remove object and gripper trajectories from ChronoFlow, respectively. **w/o CF-Sampling** disables stochastic subsampling of object keypoints during training. **w/o CF-Aug** removes ChronoFlow data augmentation used during training.

ChronoFlow trajectories and point clouds. Third, We disable this augmentation in *w/o CF-Aug*, causing Stage II success to drop from 87% to 6%, suggesting that spatial jittering is critical for stable real-world manipulation.

5 Discussion and Limitations

ChronoFlow represents temporally consistent object–gripper interactions with sparse 3D trajectories and jointly optimizes flow prediction and action generation, following the world-action modeling principle of learning future dynamics together with control. Although trajectories extracted by segmentation and 3D tracking contain noise, drift, and missing keypoints, we train with the same imperfect pipeline used at inference, together with random keypoint sampling, dropout, and history truncation. This encourages the policy to capture stable motion trends rather than rely on individual tracks. However, severe occlusion or persistent tracking failures may still degrade performance, motivating more robust and uncertainty-aware interaction tracking.

Acknowledgments

This work was supported by the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China; National Natural Science Foundation of China (No. 62506232); Science and Technology Major Project of Jiangsu Province (No. BG2024041); Shanghai Committee of Science and Technology (Yangfan: No. 24YF2722000; No. 24511103200).

Lixin Yang is the corresponding author and is affiliated with the School of Artificial Intelligence, Shanghai Jiao Tong University, Shanghai, China, 200230.

References

1. Bharadhwaj, H., Mottaghi, R., Gupta, A., Tulsiani, S.: Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In: European Conference on Computer Vision. pp. 306–324. Springer (2024)
2. Chen, B., Monso, D.M., Du, Y., Simchowitz, M., Tedrake, R., Sitzmann, V.: Diffusion forcing: Next-token prediction meets full-sequence diffusion. In: NeurIPS. pp. 24081–24125 (2024)
3. Chen, J., Fang, H., Wang, C., Wang, S., Lu, C.: History-aware visuomotor policy learning via point tracking. arXiv preprint arXiv:2509.17141 (2025)
4. Chen, T., Chen, Z., Chen, B., Cai, Z., Liu, Y., Li, Z., Liang, Q., Lin, X., Ge, Y., Gu, Z., et al.: Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. arXiv preprint arXiv:2506.18088 (2025)
5. Eisner, B., Zhang, H., Held, D.: Flowbot3d: Learning 3d articulation flow to manipulate articulated objects. In: Robotics: Science and Systems (2022)
6. Fang, H.S., Fang, H., Tang, Z., Liu, J., Wang, C., Wang, J., Zhu, H., Lu, C.: RH20T: A comprehensive robotic dataset for learning diverse skills in one-shot. In: IEEE International Conference on Robotics and Automation. pp. 653–660. IEEE (2024)
7. Fang, H., Grotz, M., Pumacay, W., Wang, Y.R., Fox, D., Krishna, R., Duan, J.: Sam2act: Integrating visual foundation model with a memory architecture for robotic manipulation. In: International Conference on Machine Learning (2025)
8. Gkanatsios, N., Xu, J., Bronars, M., Mousavian, A., Ke, T.W., Fragkiadaki, K.: 3d flowmatch actor: Unified 3d policy for single-and dual-arm manipulation. arXiv preprint arXiv:2508.11002 (2025)
9. Hsu, C.C., Wen, B., Xu, J., Narang, Y., Wang, X., Zhu, Y., Biswas, J., Birchfield, S.: Spot: Se(3) pose trajectory diffusion for object-centric manipulation. In: IEEE International Conference on Robotics and Automation. pp. 4853–4860 (2025)
10. Hu, Y., Guo, Y., Wang, P., Chen, X., Wang, Y.J., Zhang, J., Sreenath, K., Lu, C., Chen, J.: Video prediction policy: A generalist robot policy with predictive visual representations. In: International Conference on Machine Learning. pp. 24328–24346 (2025)
11. Huang, W., Chao, Y.W., Mousavian, A., Liu, M.Y., Fox, D., Mo, K., Fei-Fei, L.: Pointworld: Scaling 3d world models for in-the-wild robotic manipulation. arXiv preprint arXiv:2601.03782 (2026)
12. Huang, Z., Shi, X., Zhang, C., Wang, Q., Cheung, K.C., Qin, H., Dai, J., Li, H.: Flowformer: A transformer architecture for optical flow. In: European conference on computer vision. pp. 668–685. Springer (2022)
13. Jin, Y., Sun, Z., Xu, K., Chen, L., Jiang, H., Huang, Q., Song, C., Liu, Y., Zhang, D., Song, Y., et al.: Video-lavit: unified video-language pre-training with decoupled visual-motional tokenization. In: International Conference on Machine Learning. pp. 22185–22209 (2024)
14. Li, H., Yang, S., Chen, Y., Chen, X., Yang, X., Tian, Y., Wang, H., Wang, T., Lin, D., Zhao, F., et al.: Cronusvla: Towards efficient and robust manipulation via multi-frame vision-language-action modeling. arXiv preprint arXiv:2506.19816 (2025)
15. Li, L., Zhang, Q., Luo, Y., Yang, S., Wang, R., Han, F., Yu, M., Gao, Z., Xue, N., Zhu, X., Shen, Y., Xu, Y.: Causal world modeling for robot control. arXiv preprint arXiv:2601.21998 (2026)

16. Noh, S., Nam, D., Kim, K., Lee, G., Yu, Y., Kang, R., Lee, K.: 3d flow diffusion policy: Visuomotor policy learning via generating flow in 3d space. arXiv preprint arXiv:2509.18676 (2025)
17. Pai, J., Achenbach, L., Montesinos, V., Forrai, B., Mees, O., Nava, E.: mimic-video: Video-action models for generalizable robot control beyond vlas. arXiv preprint arXiv:2512.15692 (2025)
18. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023)
19. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., et al.: Sam 2: Segment anything in images and videos. In: International Conference on Learning Representations (2025)
20. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 234–241. Springer (2015)
21. Shi, H., Xie, B., Liu, Y., Sun, L., Liu, F., Wang, T., Zhou, E., Fan, H., Zhang, X., Huang, G.: Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. arXiv preprint arXiv:2508.19236 (2025)
22. Shi, H., Xie, B., Liu, Y., Sun, L., Liu, F., Wang, T., Zhou, E., Fan, H., Zhang, X., Huang, G.: Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. arXiv preprint arXiv:2508.19236 (2025)
23. Shi, X., Huang, Z., Li, D., Zhang, M., Cheung, K.C., See, S., Qin, H., Dai, J., Li, H.: Flowformer++: Masked cost volume autoencoding for pretraining optical flow estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1599–1610 (2023)
24. Su, Y., Liu, N., Chen, D., Zhao, Z., Wu, K., Li, M., Xu, Z., Che, Z., Tang, J.: Freqpolicy: Efficient flow-based visuomotor policy via frequency consistency. arXiv preprint arXiv:2506.08822 (2025)
25. Su, Y., Zhan, X., Fang, H., Li, Y., Lu, C., Yang, L.: Motion before action: Diffusing object motion as manipulation condition. IEEE Robotics and Automation Letters **10**(7), 7428–7435 (2025)
26. Tang, X., Qiu, J., Xie, L., Tian, Y., Jiao, J., Ye, Q.: Adaptive keyframe sampling for long video understanding. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 29118–29128 (2025)
27. Torne, M., Pertsch, K., Walke, H., Vedder, K., Nair, S., Ichter, B., Ren, A.Z., Wang, H., Tang, J., Stachowicz, K., Dhabalia, K., Equi, M., Vuong, Q., Springenberg, J.T., Levine, S., Finn, C., Driess, D.: Mem: Multi-scale embodied memory for vision language action models (2026)
28. Torne, M., Tang, A., Liu, Y., Finn, C.: Learning long-context diffusion policies via past-token prediction. In: CoRL (2025)
29. Wang, C., Fang, H., Fang, H.S., Lu, C.: Rise: 3d perception makes real-world robot imitation simple and effective. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 2870–2877 (2024)
30. Wang, X., Wang, C., Xu, Y., Ye, M., Zhang, F., Tian, J., Zhan, X., Zhu, L., Lu, C., Yang, L.: LaMP: Learning vision-language-action policy with 3d scene flow as latent motion prior. In: European Conference on Computer Vision (ECCV) (2026)
31. Wen, C., Lin, X., So, J.I.R., Chen, K., Dou, Q., Gao, Y., Abbeel, P.: Any-point trajectory modeling for policy learning. In: Robotics: Science and Systems (2024)
32. Xiao, Y., Wang, J., Xue, N., Karaev, N., Makarov, Y., Kang, B., Zhu, X., Bao, H., Shen, Y., Zhou, X.: Spatialtrackerv2: Advancing 3d point tracking with explicit camera motion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6726–6737 (2025)

33. Xu, M., Xu, Z., Xu, Y., Chi, C., Wetzstein, G., Veloso, M., Song, S.: Flow as the cross-domain manipulation interface. In: Conference on Robot Learning. vol. 270, pp. 2475–2499. PMLR (2024)
34. Xu, M., Gao, M., Li, S., Lu, J., Gan, Z., Lai, Z., Cao, M., Kang, K., Yang, Y., Dehghan, A.: Slowfast-llava-1.5: A family of token-efficient video large language models for long-form video understanding. In: Conference on Language Modeling (2025)
35. Ye, S., Ge, Y., Zheng, K., Gao, S., Yu, S., Kurian, G., Indupuru, S., Tan, Y.L., Zhu, C., Xiang, J., et al.: World action models are zero-shot policies. arXiv preprint arXiv:2602.15922 (2026)
36. Yu, S., Jin, C., Wang, H., Chen, Z., Jin, S., Zuo, Z., Xu, X., Sun, Z., Zhang, B., Wu, J., et al.: Frame-voyager: Learning to query frames for video large language models. In: International Conference on Learning Representations (2025)
37. Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., Levine, S.: Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In: Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 100, pp. 1094–1100. PMLR (2019)
38. Ze, Y., Zhang, G., Zhang, K., Hu, C., Wang, M., Xu, H.: 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In: Robotics: Science and Systems (2024)
39. Zhang, B., Ke, L., Harley, A.W., Fragkiadaki, K.: Tapip3d: Tracking any point in persistent 3d geometry. arXiv preprint arXiv:2504.14717 (2025)
40. Zhang, C., Le Moing, G., Koppula, S., Rocco, I., Momeni, L., Xie, J., Sun, S., Sukthankar, R., Barral, J.K., Hadsell, R., Ghahramani, Z., Zisserman, A., Zhang, J., Sajjadi, M.S.M.: Efficiently reconstructing dynamic scenes one d4rt at a time. In: CVPR (2026)
41. Zheng, R., Liang, Y., Huang, S., Gao, J., Daumé III, H., Kolobov, A., Huang, F., Yang, J.: Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. In: International Conference on Learning Representations (2025)
42. Zhu, C., Yu, R., Feng, S., Burchfiel, B., Shah, P., Gupta, A.: Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. In: Proceedings of Robotics: Science and Systems (RSS) (2025)

Appendix

A ChronoFlow Implementation Details

This section details the construction of **ChronoFlow**, including how gripper and object keypoints are defined, how their temporal flows are obtained, and how historical and future ChronoFlow trajectories are organized.

ChronoFlow keypoints. ChronoFlow represents the gripper-object interaction state at each timestep t using sparse 3D keypoints defined on both the robot gripper and task-relevant objects. We denote the gripper keypoints as $\boldsymbol{P}_t^g \in \mathbb{R}^{N_g \times 3}$ and the object keypoints as $\boldsymbol{P}_t^o \in \mathbb{R}^{N_o \times 3}$, where N_g and N_o are the numbers of gripper and object keypoints, respectively. Their union $\boldsymbol{P}_t = \boldsymbol{P}_t^g \cup \boldsymbol{P}_t^o$ forms a compact abstraction of the physical interaction state.

Gripper keypoints flow. We predefine N_g canonical keypoints on the gripper geometry, rigidly attached to the tool center point (TCP). Given the TCP pose at each timestep, we compute the 3D positions of all gripper keypoints via rigid transformation, producing temporally consistent gripper trajectories in the world coordinate frame.

Object keypoints flow. For manipulated objects, we segment task-relevant objects from the initial observation, e.g., using 3D SAM-2 [19], and apply Farthest Point Sampling (FPS) to obtain candidate keypoints on each object. We then track these points over time using a 3D point tracker, e.g., TAPIP3D [39], to obtain temporally consistent object-centric trajectories. In practice, instead of using all tracked candidates, we randomly sample N_o object keypoints per sequence for training, which reduces overfitting to specific spatial locations and improves generalization across interaction configurations.

Historical ChronoFlow trajectory. Given a history horizon h_p, we construct the historical ChronoFlow trajectory as $\boldsymbol{P}_{t-h_p:t-1} = \{\boldsymbol{P}_{t-h_p}, \dots, \boldsymbol{P}_{t-1}\}$, where $\boldsymbol{P}^g_{t-h_p:t-1} \in \mathbb{R}^{h_p \times N_g \times 3}$ and $\boldsymbol{P}^o_{t-h_p:t-1} \in \mathbb{R}^{h_p \times N_o \times 3}$. It contains both gripper and object keypoint flows over the past window. For conciseness, we use $\boldsymbol{P}_{:t} = \{\boldsymbol{P}_\tau\}_{\tau<t}$ to denote the historical ChronoFlow trajectories constructed from past observations.

Future ChronoFlow trajectory. Following the notation in the main paper, the policy predicts a horizon-length ChronoFlow trajectory $\boldsymbol{P}_{t:t+H}$, which describes the future evolution of gripper-object interactions starting from timestep t. Its gripper and object components are denoted as $\boldsymbol{P}^g_{t:t+H} \in \mathbb{R}^{H \times N_g \times 3}$ and $\boldsymbol{P}^o_{t:t+H} \in \mathbb{R}^{H \times N_o \times 3}$, respectively. Equivalently, the full trajectory is represented as $\boldsymbol{P}_{t:t+H} \in \mathbb{R}^{(N_g+N_o) \times H \times 3}$.

B ChronoFlow Policy Implementation Details

This section provides implementation details of **ChronoFlow Policy**, including the current observation encoder, and the conditioning mechanisms used in different diffusion backbones, i.e., Unet1D [20] and DiT [18].

B.1 Current Observation Encoder

Observation. ChronoFlow Policy conditions on the current observation $\boldsymbol{o}_t = (\boldsymbol{p}_t, \boldsymbol{q}_t)$ at decision time t. Here $\boldsymbol{p}_t \in \mathbb{R}^{N \times 6}$ denotes the scene-level point cloud, where each point contains 3D coordinates and RGB values. When available, $\boldsymbol{q}_t \in \mathbb{R}^{d_q}$ denotes the proprioceptive state, such as joint positions or end-effector pose. The observation encoder maps $\boldsymbol{o}_t$ to a global feature vector $\boldsymbol{c}_t = E_{\text{obs}}(\boldsymbol{o}_t) \in \mathbb{R}^{D_{\text{obs}}}$, which conditions the diffusion backbone.

Point cloud branch. We use different point-cloud encoders in simulation and real-world settings. In simulation, we adopt a PointNet-style point-cloud encoder composed of shared per-point MLP layers with widths `[64, 128, 256, 512]`, followed by symmetric max pooling over points and a final linear projection to produce the compact point-cloud embedding $\boldsymbol{z}_t^{\mathrm{pc}} \in \mathbb{R}^{D_{\mathrm{pc}}}$. In real-robot experiments, the raw point cloud is noisier and contains substantially more points than in simulation. Therefore, we use a sparse 3D encoder to efficiently extract robust point features from dense point clouds.

State branch. When proprioception is available, we encode $\boldsymbol{q}_t \in \mathbb{R}^{d_q}$ using a lightweight MLP with ReLU activations. In all simulation experiments, we set `state_mlp_size=(64, 64)`, yielding a two-layer MLP $d_q \to 64 \to 64$ and producing a state embedding $\boldsymbol{z}_t^q \in \mathbb{R}^{64}$. In real-robot experiments, proprioceptive state is omitted and the observation feature is extracted from the point cloud only.

Observation feature fusion. When both point-cloud and proprioceptive features are available, we concatenate $\boldsymbol{z}_t^{\mathrm{pc}}$ and $\boldsymbol{z}_t^q$ along the channel dimension and apply a final linear layer. The resulting feature $\boldsymbol{c}_t \in \mathbb{R}^{D_{\mathrm{obs}}}$ is used as the observation condition for ChronoFlow diffusion. We set $D_{\mathrm{obs}} = 128$ in our implementation.

B.2 Conditioning and Diffusion Backbone

We implement ChronoFlow Policy with diffusion backbones, including Unet1D [20] and DiT [18]. Since our simulation benchmarks are approximately Markovian while real-world manipulation exhibits stronger non-Markovian dependencies, we use different conditioning signals in these two settings.

ChronoFlow diffusion target. At each timestep t, ChronoFlow represents the interaction state as $\boldsymbol{P}_t = \boldsymbol{P}_t^g \cup \boldsymbol{P}_t^o \in \mathbb{R}^{N\times 3}$, where $N = N_g + N_o$. For a prediction horizon H, the diffusion model predicts the ChronoFlow trajectory $\boldsymbol{P}_{t:t+H} \in \mathbb{R}^{N\times H\times 3}$. We apply the standard forward diffusion process to $\boldsymbol{P}_{t:t+H}$ and denote the noisy trajectory at diffusion step k as $\boldsymbol{P}_{t:t+H}^k$. The denoising network predicts the diffusion target, e.g., noise ϵ under the DDPM objective, conditioned on the current observation and historical ChronoFlow.

Simulation setting. In simulation, tasks are approximately Markovian and the current interaction state is sufficient for decision making. Therefore, we inject the current ChronoFlow state $\boldsymbol{P}_t$ as an explicit trajectory condition. Specifically, we repeat $\boldsymbol{P}_t$ along the prediction horizon to obtain $\widetilde{\boldsymbol{P}}_{t:t+H} = \textsc{Repeat}(\boldsymbol{P}_t)$ and concatenate it with the noisy ChronoFlow trajectory:

$$\boldsymbol{S}_{t:t+H}^k = \left[\widetilde{\boldsymbol{P}}_{t:t+H}; \boldsymbol{P}_{t:t+H}^k\right] \in \mathbb{R}^{H\times N\times 6}. \tag{9}$$

The diffusion backbone takes $\boldsymbol{S}_{t:t+H}^k$, the diffusion timestep k, and the observation feature $\boldsymbol{c}_t = E_{\mathrm{obs}}(\boldsymbol{o}_t)$ as input, and predicts the denoising target for the noisy ChronoFlow trajectory.

Real-world setting. In real-world long-horizon manipulation, we run the 3D point tracker asynchronously with the policy for efficiency. As a result, the policy may not have access to the up-to-date current ChronoFlow state $\boldsymbol{P}_t$ at inference time. Therefore, in real-robot experiments, we do not use $\boldsymbol{P}_t$ as an explicit repeated trajectory condition. Instead, the model denoises the future ChronoFlow trajectory from Gaussian noise while conditioning on both the historical ChronoFlow $\boldsymbol{P}_{:t}$ and the current point-cloud observation feature.

Concretely, at each diffusion step k, the ChronoFlow encoder jointly encodes $\boldsymbol{P}_{:t}$ and $\boldsymbol{P}^k_{t:t+H}$ into tokens $\boldsymbol{Z}^k_t$. The observation encoder extracts the current point-cloud feature $\boldsymbol{c}_t = E_{\text{obs}}(\boldsymbol{p}_t)$. The diffusion backbone then uses $\boldsymbol{Z}^k_t$, $\boldsymbol{c}_t$, and the diffusion timestep k as conditioning signals to perform iterative denoising.

Action decoding. After iterative denoising, the model obtains clean ChronoFlow tokens $\boldsymbol{Z}^0_t$ and decodes them into the predicted future interaction trajectory $\hat{\boldsymbol{P}}_{t:t+H} = D_{\text{CF}}(\boldsymbol{Z}^0_t)$. The action decoder predicts future robot actions from the clean tokens and the decoded ChronoFlow trajectory:

$$\hat{\boldsymbol{a}}_{t:t+H} = \mathcal{A}_\phi\left(\left[\boldsymbol{Z}^0_t; \hat{\boldsymbol{P}}_{t:t+H}\right]\right). \tag{10}$$

During training, action prediction is supervised with an MSE objective, while the diffusion backbone is trained with ChronoFlow denoising supervision.

C Baseline Implementation Details

This section summarizes implementation details of the baselines and highlights their differences from **ChronoFlow Policy**. To ensure a fair comparison, all baselines use the same point-cloud observation encoder, normalization, diffusion scheduler, and shared module hyperparameters whenever applicable. We only vary the core modeling choices, including history usage, interaction representation, prediction targets, and training objectives.

C.1 RISE

RISE [29] is an *action-only* diffusion policy. **Key differences to ChronoFlow Policy:** it does not predict interaction trajectories, does not perform keypoint/flow diffusion, and does not incorporate an explicit ChronoFlow encoder. Given a single point-cloud observation at time t, RISE first encodes the point cloud to obtain point features, aggregates them with a Transformer readout, and conditions an action diffusion head on the resulting global feature.

Shared observation encoder. RISE uses the same point-cloud observation encoder as ChronoFlow Policy to extract point features from the current point cloud.

Transformer aggregation. Point features are processed by a Transformer with `hidden_dim` $= 512$, `nheads` $= 8$, `n_encoder_layers` $= 4$, `n_decoder_layers` $= 1$, `dim_feedforward` $= 2048$, and dropout 0.1. A learnable readout token is appended, and the readout feature $\boldsymbol{r}_t \in \mathbb{R}^{512}$ is used as the global condition.

Action diffusion head. The action head is a conditional Unet1D diffusion model, implemented as `DiffusionUNetPolicy`. It predicts `num_action` $= 20$ action steps with per-step dimension `action_dim` $= 10$. RISE is trained and evaluated with **action diffusion only**.

C.2 HistRISE

HistRISE [3] is an *action-only* diffusion policy that augments RISE with point-track history features. **Key differences to ChronoFlow Policy:** HistRISE does not predict future interaction trajectories, has no ChronoFlow supervision, and does not perform diffusion over interaction-centric keypoint representations. It only trains an action diffusion head.

Shared observation encoder. HistRISE uses the same point-cloud observation encoder as ChronoFlow Policy and only differs in the additional history track features.

History point tracks and track encoder. HistRISE takes normalized point tracks as history input, where `num_targets` $= 1$ and `points_per_target` $= 5$ by default. For fair comparison, we use the same track encoder design as ChronoFlow Policy for extracting history features: temporal patch embedding with `patch_size` $= 4$ and `embed_dim` $= 256$, followed by cross-attention pooling with a learnable query. The cross-attention module uses `query_dim` $= 512$, `num_queries` $= 1$, `n_layers` $= 4$, `n_heads` $= 8$, `ff_dim` $= 1024$, dropout 0.1, and time embedding.

Token fusion. The encoded track features are projected to the Transformer hidden dimension and appended to the point feature sequence before Transformer readout. The readout feature conditions the same action diffusion head as RISE. HistRISE is trained and evaluated with **action diffusion only**.

C.3 Motion Before Action (MBA)

MBA [25] uses a *two-stage* diffusion pipeline in which an object-motion summary is first sampled and then used to condition action diffusion. **Key differences to ChronoFlow Policy:** MBA does not model interaction-centric gripper-object keypoints. Instead, it predicts object pose trajectories and does not use an explicit ChronoFlow history encoder.

Shared observation encoder. MBA uses the same observation encoder as ChronoFlow Policy to extract the current observation feature from the point cloud.

Two-stage diffusion. MBA first samples a future object-motion trajectory, stored as `obj_pos` with dimension $d_o = 6$, using a conditional Unet1D diffusion model. It then generates actions using a second conditional Unet1D diffusion model conditioned on the predicted object motion.

Condition construction. For action generation, MBA embeds the predicted object-motion trajectory with an MLP of width `[32,32]` and concatenates it with the current observation feature, followed by a final projection to a 128-dimensional global condition.

Diffusion backbone. Both diffusion stages use `ConditionalUnet1D` with `down_dims` $= (256, 512, 1024)$, diffusion embedding dimension 256, kernel size 5, group norm groups 8, and FiLM conditioning by default.

C.4 3D Flow Diffusion Policy (3D-FDP)

3D-FDP [16] is a two-stage diffusion policy that first predicts a future *scene-level* point-cloud trajectory and then generates actions conditioned on the predicted future scene. **Key differences to ChronoFlow Policy:** 3D-FDP predicts future dynamics at the entire-scene point-cloud level rather than using interaction-centric gripper-object keypoints, and it does not incorporate ChronoFlow history.

Shared observation encoder. 3D-FDP uses the same observation encoder as ChronoFlow Policy to extract the current observation feature.

Stage I: future point-cloud diffusion. 3D-FDP predicts a future point-cloud sequence over horizon H with N_{PC} scene points by diffusing flattened XYZ coordinates:

$$\boldsymbol{X}^{\text{FUT}}_{t:t+H} \in \mathbb{R}^{H\times(N_{\text{PC}}\cdot 3)}. \tag{11}$$

This stage conditions only on the current observation feature and does not use any history encoder.

Stage II: action diffusion conditioned on predicted scene feature. After sampling the future point cloud, 3D-FDP encodes it into a compact feature using a PointNet-style encoder with output dimension `pred_pc_feature_dim` $= 64$ and averages per-timestep features over the horizon. The action diffusion model conditions on the concatenation of the observation feature and the predicted future-scene feature, with dimensions $128 + 64$ by default.

Diffusion backbone. Both point-cloud diffusion and action diffusion use `Unet1D` with `down_dims` $= (256, 512, 1024)$, diffusion embedding dimension 256, kernel size 5, group norm groups 8, and FiLM conditioning by default. The point-cloud diffusion loss is weighted by $\lambda_{\text{PC}} = 1$.

D Training Curves

To further analyze the learning behavior of different policies, we report training curves on representative tasks from MetaWorld and RoboTwin. All methods follow the same evaluation protocol as in the main paper. These curves complement the final success-rate comparisons by showing how quickly each method improves during training and how stable the learned policy is across evaluation checkpoints.

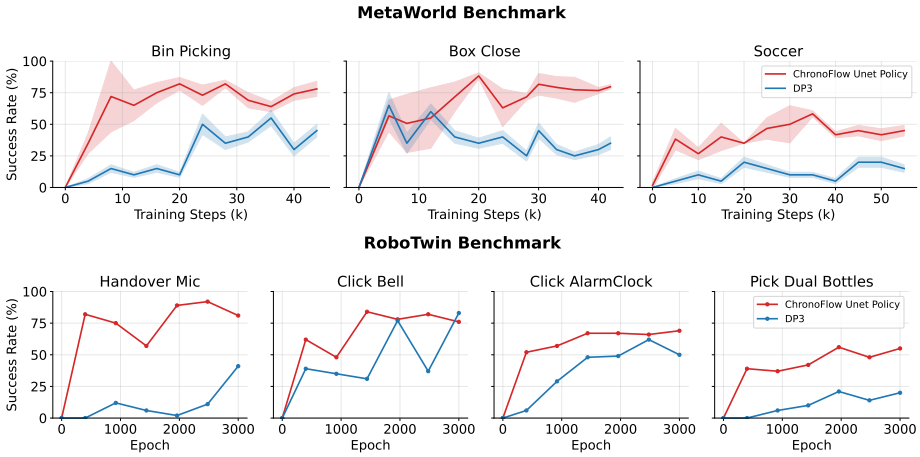


Fig. 5: Learning curves on simulation benchmarks. CFP (Unet) generally converges faster and achieves higher or comparable final success rates than DP3 across representative tasks, demonstrating the effectiveness of ChronoFlow supervision.

As shown in Fig. 5, CFP achieves faster convergence than DP3 on both single-arm and dual-arm manipulation tasks. On MetaWorld, CFP rapidly improves on interaction-intensive tasks such as *Bin Picking* and maintains a clear advantage throughout training. On RoboTwin, CFP reaches high success rates earlier on long-horizon or coordination-heavy tasks such as *Handover Mic* and *Pick Dual Bottles*. These results indicate that ChronoFlow provides a stronger training signal than action-only supervision, helping the policy learn object–gripper interaction dynamics more efficiently.