STAMP: Spatio-Temporal Augmented Memory Policy for Robotic Manipulation

Zhirun Yue[1,3], Mingxin Wang[1], Tianyi You[2], Jun Cheng[3], Houde Liu[1,*]
[1]Shenzhen International Graduate School, Tsinghua University, Shenzhen, China
[2]College of Electrical Engineering and Control Science, Nanjing Tech University, Nanjing, China
[3]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
yuezr23@mails.tsinghua.edu.cn, wmx25@mails.tsinghua.edu.cn, youtianyi@njtech.edu.cn, jun.cheng@siat.ac.cn
[*]Corresponding author: liu.hd@sz.tsinghua.edu.cn

***Abstract*—Efficient robotic manipulation via imitation learning aims to distill complex skills from expert demonstrations. Acquiring skills for complex manipulation tasks necessitates a consistent understanding of both spatial configurations and temporal dynamics. While conditional information guides action generation in current models, the absence of explicit 4D world modeling leads to a misalignment between environmental perception and execution, hindering the temporal consistency of generated behaviors. Driven by this, we introduce STAMP (Spatio-Temporal Augmented Memory Policy), a novel architecture that embeds inherent spatiotemporal reasoning directly into the policy's decision-making process. Specifically, STAMP performs temporal-aware feature extraction on both observations and actions to capture their co-evolution. Inspired by human cognitive science, we design a hierarchical memory pyramid that represents historical context across multi-level granularities, enabling the policy to model the evolution of memory over time in a way that mimics biological information consolidation. Extensive evaluations on Adroit and MetaWorld demonstrate that STAMP achieves success rates with a 2.4% improvement, while maintaining high efficiency for real-time control.**

***Index Terms*—robot learning, robotic manipulation, flow matching, embodied ai, memory module**

I. INTRODUCTION

Artificial intelligence technology has greatly empowered robotics and promoted the advancement of embodied AI [1], [2]. Enabling robots equipped with various sensory inputs to master diverse manipulation skills in real world has been a long-standing objective in the robotic community [3], [4]. However, many existing researches rely primarily on 2D perception and underutilize 3D spatial information, which carries critical semantic and geometric cues about objects and scenes [5], [6]. Moreover, a large fraction of visuomotor policies generate actions based only on the current observation, without explicitly modeling temporal knowledge about world dynamics, which limits their ability to reason about the behavior pattern and degrades action quality in complex manipulation tasks [7].

Recently, most visuomotor systems encode 2D image that provide only a projection of inherently three-dimensional scenes [8]. Such 2D representations struggle to capture fine-grained geometry, leading to brittle spatial reasoning under partial conclusions and contact-rich interactions [9]. In contrast, geometry-grounded visual foundation models, such as PointNet++, further strengthen spatial perception capability by learning 3D-aware features, offering a more robust and generalizable visual backbone for manipulation [10], [11].

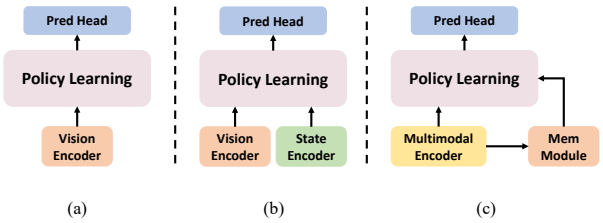

Fig. 1: Paradigms of action generative models: (a) vision-only policy learning; (b) multimodal-fusion policy learning; and (c) memory-augmented policy learning.

Robotic manipulation is fundamentally a spatiotemporal problem: the correct action often depends not only on the current observation, but also on how the scene and the robot state have evolved over time [12]. This motivates the need for memory-aware spatial modeling [13]. Fig 1 summarizes three representative paradigms of action generative models. By maintaining an intrinsic memory of past 3D observations alongside proprioceptive–action history, the policy can construct a consistent representation of world dynamics, recover information lost to geometric constraints, and reason about long-horizon dependencies such as phase transitions and contact evolution [14].

Generative modeling has emerged as a pervasive and effective framework for action synthesis in robotic manipulation [15], [16]. While diffusion-based policies are widely adopted for their ability to model complex action distributions [17], [18], Flow Matching (FM) offers a more efficient alternative by providing a more direct ODE-based transport from noise to actions with fewer inference steps [19]. In particular, Consistency Flow Matching enables near one-step generation via a straight-line flow, making it attractive for real-time control [20], [21]. Meanwhile, embodied manipulation is

This work was supported by the Shenzhen Science and Technology Program (Grant No. RCJC20210706091946001ZDCY20250901104207008)

inherently spatiotemporal. Observations, proprioception, and actions evolve continuously, which makes short-term histories crucial for disambiguating 3D geometry and task phases that are invisible in a single frame. However, integrating temporal memory into Consistency-FM for manipulation, in the sense of learning a 4D manipulation policy conditioned on spatiotemporal context, remains an underexplored area in robot policy learning.

In this paper, we propose the **Spatio-Temporal Augmented Memory Policy (STAMP)**, a novel flow-matching method for contact-rich manipulation to address the challenges of visual understanding and contextual cognition. Instead of conditioning on a single observation, our policy summarizes multi-step temporal context by a pyramid-style temporal memory module that progressively fuses historical observations into multi-scale representations, and then gates these memories with the current observation to obtain a context-aware state embedding. To capture actuator dynamics and disambiguate short-term intent, we additionally encode a short horizon of action history as priors fused with the observation memory through a lightweight conditioning network, forming a compact condition for action generation. Conditioned on this temporally grounded representation, we train a consistency flow-matching model to generate future action sequences directly from pure noise, aligning training and inference and enabling efficient sampling with very few steps. This design explicitly leverages temporal continuity and interaction phase cues, yielding more stable and coherent behavior than observation-only methods in long-horizon manipulation. In summary, our main contribution are categorized into three aspects:

- We propose a unified multimodal conditional policy that captures complementary cues from state–vision–action evolution for robust manipulation learning.
- We introduce a temporally-aware memory pyramid module that stores and aggregates long-term historical features, augmented with temporal embeddings to preserve time ordering. This enables effective utilization of long-horizon context to alleviate memory collapse and facilitates a better comprehension of the world state.
- We design a 4D spatiotemporal modeling framework that enables robust robotic manipulation by incorporating a structured memory update mechanism spanning feature processing, memory consolidation, and context retrieval

II. RELATED WORK

A. Generative Policies for Manipulation

In the field of embodied AI, robot manipulation learning has witnessed extensive exploration, with the end-to-end learning paradigm emerging as the prevailing approach [22], [23]. Current models predominantly employ generative architectures to predict future action chunks, benefiting from their inherent ability to capture complex behavioral patterns from raw observations [24]. Diffusion Policy (DP) formulates robot visuomotor control as an image-conditioned denoising process by iteratively refining high-dimensional action trajectories from Gaussian noise [6]. To better capture the underlying 3D structure of the physical world, 3D Diffusion Policy (DP3) injects explicit spatial and geometric structure by conditioning the diffusion action generator on a compact 3D representation extracted from sparse point clouds, enabling more generalizable visuomotor control [5]. Recent VLA-style manipulation policies increasingly prioritize explicit spatial structures to enhance performance [25], [26]. However, such large-scale vision encoders are not lightweight and thus incur substantial computation, limiting real-time deployment. Flow matching is a generative model that directly regresses a continuous-time probability flow, typically enabling faster sampling than diffusion [27]. Consistency Policy shows that enforcing self-consistency along a diffusion teacher's trajectories can distill slow multi-step denoising into low-latency visuomotor control, and Consistency Flow Matching (CFM) generalizes this idea with a principled velocity-consistency constraint that directly defines straighter ODE probability flows for faster generation [28]. Prior studies have demonstrated the efficacy of Flow Matching (FM) across diverse robotic applications, including inverse reinforcement learning [29], multi-support motion planning [30], and real-world robot deployment [31]. ManiCM relies on consistency distillation for one-step generation, whereas CFM utilizes a more streamlined, distillation-free training objective [32]. The efficacy of this direct approach for robot manipulation has yet to be explored.

B. Temporal Modeling in Robot Policies

Drawing inspiration from human cognitive processes, RoboMemory leverages brain-inspired multi-memory architectures to empower embodied AI agents with a deeper understanding of the world [13]. To address memory collapse, MemoryVLA proposes a decoupled perceptual-cognitive memory architecture that disentangles instantaneous sensory processing from long-term knowledge retrieval, ensuring that historical context effectively guides long-horizon manipulation [7]. While RoboFlamingo leverages an LSTM [33] to propagate latent vision-language tokens, the inherent coarseness of this representation leads to the loss of fine-grained spatiotemporal details that are crucial for complex manipulation [34]. CoT-VLA introduces visual chain-of-thought by autoregressively predicting future image-frame tokens as intermediate visual goals before generating a short action-token chunk [35]. The world model paradigm has emerged as a pivotal framework for enhancing robot perception, enabling more rigorous modeling of spatial and temporal consistency across complex interaction sequences [36], [37]. CogVLA improves cognitive depth by repeatedly injecting task-relevant cues via dynamic routing [38]. It relies on explicit feature alignment rather than training an implicit, parameterized memory system, thereby constraining its generalization to complex, novel environments.

III. PRELIMINARY

Imitation learning (IL) has emerged as a fundamental paradigm for robotic manipulation, enabling agents to acquire complex skills by leveraging expert demonstrations. Let $\mathcal{A}$

denote the action space and $\mathcal{O}$ the observation space which comprises both proprioceptive robot states and visual observations; formally, $\mathcal{O} = \mathcal{X} \times \mathcal{V}$ with $o_t = (x_t, v_t)$. The policy π aims to learn the underlying expert distribution by mapping historical context to a sequence of future decisions. The probabilistic objective is defined as follows:

$$a_{future} \sim p(a_{future} \mid o_{history}, a_{history}), \tag{1}$$

where $o \in \mathcal{O}$, $a \in \mathcal{A}$ denote the observations and actions, respectively.

Specifically, given a fixed-size observation history window k_o and action history window k_a, the model takes a sequence of observations $\mathbf{O}_t = \{o_i\}_{i=t-k_o+1}^{t}$ and actions $\mathbf{A}_{t-1} = \{a_i\}_{i=t-k_a}^{t-1}$ as input. The objective is to optimize the network parameters θ to regress a sequence of future actions over a prediction horizon h:

$$\min_{\theta} \mathbb{E}_{\tau \sim \mathcal{D}} \left[\mathcal{L} \left(f_\theta(\mathbf{O}_t, \mathbf{A}_{t-1}), \{a_i\}_{i=t}^{t+h-1} \right) \right], \tag{2}$$

where $\tau \sim \mathcal{D}$ denotes that the trajectories are sampled from the expert demonstration dataset $\mathcal{D}$.

IV. METHODOLOGY

In this section, we describe our STAMP in detail, the overall architecture of which is illustrated in Fig. 2. The overall architecture consists of four modules: 1) The Spatiotemporal Feature Extraction module encodes multi-frame historic spatial observation together with past actions into temporally aligned feature sequences; 2) The Memory Pyramid module applies stacked memory filters in a coarse-to-fine pyramid to progressively aggregate and store multi-granularity temporal patterns, yielding hierarchical memory representations; 3) The Adaptive Fusion module leverages a gated mechanism to adaptively fuse the current observation with the enhanced memory into a compact global condition vector; 4) The Conditional Action Generation module performs conditional action synthesis via consistency flow matching, generating the future action sequences from a short-horizon action prior while guided by the learned global condition. Additionally, we adopt a hybrid inpainting training scheme that stochastically masks future-action blocks and trains the model to recover them under the same conditional flow, while maintaining pure-noise sampling at inference.

A. Spatiotemporal Feature Extractor

1) Geometric and Proprioceptive Encoding: For the spatial modality, we employ a PointNet++ backbone to extract geometric features from the current point cloud $x_t^{\mathrm{pc}} \in \mathbb{R}^{N \times C}$, producing a vision embedding $e_t^{\mathrm{pc}} \in \mathbb{R}^{D_u}$. In parallel, a lightweight MLP encodes the proprioceptive state $x_t^{\mathrm{s}} \in \mathbb{R}^{D_s}$ to obtain $e_t^{\mathrm{s}} \in \mathbb{R}^{D_u}$. To explicitly model temporal ordering, we inject a timestep embedding $\phi(t)$ into both branches and fuse it with modality-specific embeddings via addition, yielding time-conditioned observation representations:

$$z_t^{\mathrm{pc}} = f_{pc}(x_t^{\mathrm{pc}}, \phi(t)), \quad z_t^{\mathrm{s}} = f_s(x_t^{\mathrm{s}}, \phi(t)). \tag{3}$$

The final per-step observation token z_t^{o} is formed by fusing the two modalities via a learnable fusion layer.

Algorithm 1: Memory Pyramid Hierarchy

Input: Initial memory bank $\mathcal{B}_0 = \{m_1^0, m_2^0, \ldots, m_{L_0}^0\}$
Output: Multi-level compressed memory banks $\{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_K\}$ where $K = (L_0 - 1)//2$

$\mathcal{B} \leftarrow \mathcal{B}_0$;
$K \leftarrow (L_0 - 1)//2$;
for $j \leftarrow 1$ **to** K **do**
  $\mathcal{B}_j \leftarrow \emptyset$; `// Initialize the j-th level memory bank`
  for $i \leftarrow 2$ **to** $|\mathcal{B}_{j-1}| - 1$ **do**
    $\hat{m}_i \leftarrow \text{CrossAttention}(m_i^{j-1}, m_{i-1}^{j-1})$; `// Contextual coupling`
    $m_i^j \leftarrow \text{CrossAttention}(m_{i+1}^{j-1}, \hat{m}_i)$; `// Pyramidal abstraction`
    $\mathcal{B}_j \leftarrow \mathcal{B}_j \cup \{m_i^j\}$;
  end
  $|\mathcal{B}_{j-1}| \leftarrow |\mathcal{B}_j|$;
end
return $\{\mathcal{B}_0, \ldots, \mathcal{B}_K\}$; `// Return the pyramid of memory representations`

2) Temporal Action Modeling with GRU: Inspired by the principle of local continuity in fluid physical motions, we design a dedicated GRU-based structure [39] to process the historical action sequence $\{a_{t-k_a}, \ldots, a_{t-1}\}$. Unlike static encoders, the GRU is specifically tailored to capture the long-term dependencies and causal relationships inherent in the action history window k_a:

$$prior_t^{\mathrm{a}} = f_{\mathrm{a}}\left(a_{t-k_a:\, t-1}\right) \in \mathbb{R}^{D_{token}}. \tag{4}$$

The final hidden state of the GRUs serves as a condensed temporal summary of the agent's prior behavior, which facilitates the synthesis of more precise and physically plausible trajectories, thereby elevating the overall quality of the subsequent action generation.

B. Memory Pyramid Module

Robotic manipulation requires not only reacting to the current observation but also exploiting structured temporal context that reveals task phase and interaction dynamics. To this end, we introduce a Memory Pyramid Module that hierarchically aggregates historical observations into multi-granularity memory representations. The algorithm is detailed in Algorithm 1.

1) Historical feature sequence: Let $\mathbf{o}_t \in \mathbb{R}^{D_o}$ denote the encoded observation feature at time step t produced by the spatiotemporal feature extractor. Given an observation window of length T_o, we separate the current feature $\mathbf{o}_{\text{now}}$ and the historical feature sequence

$$\mathbf{M}^{(0)} = \left[\mathbf{o}_{t-T_o+1}, \ldots, \mathbf{o}_{t-1}\right] \in \mathbb{R}^{L_0 \times D_o}, \tag{5}$$

where $L_0 = T_o - 1$ and $\mathbf{o}_{\text{now}} = \mathbf{o}_t$.

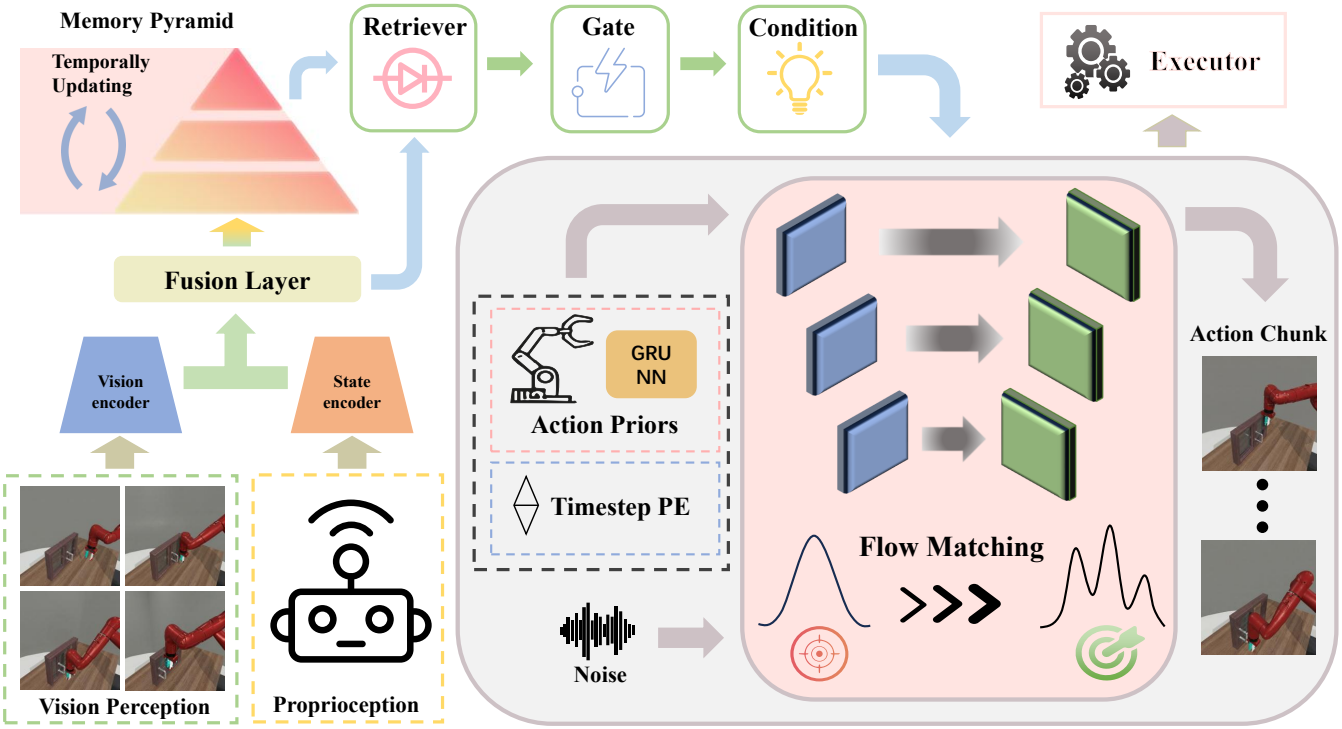

Fig. 2: The overall architecture.

2) Pyramid construction with temporal filtering: We construct a pyramid of memory banks by repeatedly applying a temporal filtering operator over triplets of adjacent features. As illustrated in Fig. 3, the architecture consists of a hierarchical arrangement of Memory Elements and Process Elements, which collaboratively facilitate the pyramidal feature abstraction process. Specifically, at pyramid level ℓ, the memory bank is a sequence $\mathbf{M}^{(\ell)} \in \mathbb{R}^{L_\ell \times D_o}$, and the next level is obtained by sliding a window of size 3:

$$\mathbf{m}_i^{(\ell+1)} = \phi\Big(\mathbf{m}_i^{(\ell)}, \mathbf{m}_{i+1}^{(\ell)}, \mathbf{m}_{i+2}^{(\ell)}\Big), \quad i = 1, \ldots, L_\ell - 2, \quad (6)$$

where $\phi(\cdot)$ is a learnable temporal filter unit (e.g., attention-based fusion [40]), producing

$$\mathbf{M}^{(\ell+1)} = \big[\mathbf{m}_1^{(\ell+1)}, \ldots, \mathbf{m}_{L_\ell-2}^{(\ell+1)}\big] \in \mathbb{R}^{(L_\ell-2)\times D_o}, \quad L_{\ell+1} = L_\ell - 2. \quad (7)$$

Repeating this operation yields a memory pyramid $\{\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(L)}\}$ with progressively deeper temporal abstraction, where the top level satisfies $L_L = 1$. Once the new observation are generated and stored in the memory pyramid, all elements across the hierarchy undergo a bottom-up update to maintain temporal consistency.

3) Multi-scale memory readout: Each memory bank captures temporal patterns at a distinct granularity. To obtain memory cues aligned with the current time, we read the most recent element from each level:

$$\mathbf{r}^{(\ell)} = \mathrm{Last}\Big(\mathbf{M}^{(\ell)}\Big) = \mathbf{m}_{L_\ell}^{(\ell)} \in \mathbb{R}^{D_o}, \quad \ell = 0, \ldots, L. \quad (8)$$

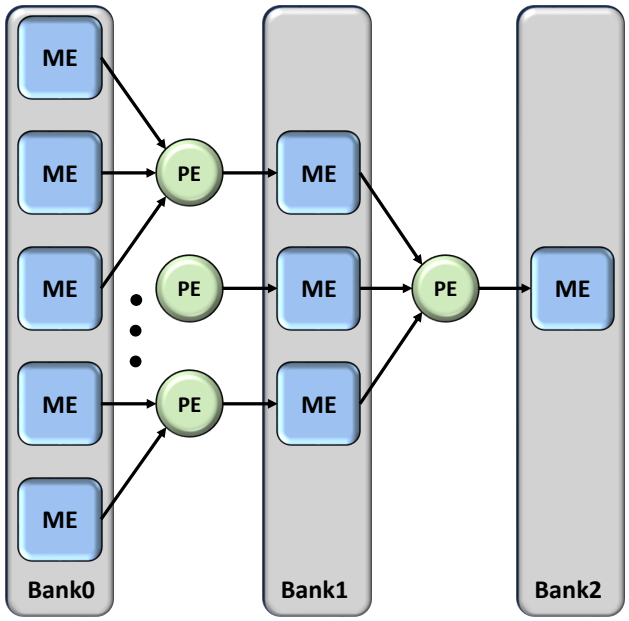

Fig. 3: The Architecture of Memory Banks.

C. Adaptive Fusion Module

1) Gated fusion with the current observation: We then adaptively fuse each multi-scale memory vector $\mathbf{r}^{(\ell)}$ with the

current observation feature $\mathbf{o}_{\text{now}}$ using a feature-wise gate:

$$\mathbf{g}^{(\ell)} = \sigma\Big(\mathrm{MLP}_g\big([\mathbf{o}_{\text{now}}, \mathbf{r}^{(\ell)}]\big)\Big) \in (0,1)^{D_o}, \tag{9}$$

$$\mathbf{f}^{(\ell)} = \mathbf{g}^{(\ell)} \odot \mathbf{o}_{\text{now}} + \big(\mathbf{1} - \mathbf{g}^{(\ell)}\big) \odot \mathbf{r}^{(\ell)} \in \mathbb{R}^{D_o}, \tag{10}$$

where $\sigma(\cdot)$ is the sigmoid function and $\odot$ denotes element-wise multiplication. This design allows the policy to dynamically balance instantaneous evidence and temporal memory at each granularity.

2) Condition vector generation: Finally, we aggregate the current feature and the fused multi-scale features into a compact global condition vector:

$$\mathbf{c}_{\text{mem}} = \mathrm{Proj}\Big([\mathbf{o}_{\text{now}}, \mathbf{f}^{(0)}, \mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(L)}]\Big) \in \mathbb{R}^{D_c}, \tag{11}$$

which is subsequently used as a global conditioning signal for conditional action generation. In practice, $\mathrm{Proj}(\cdot)$ is implemented as an MLP with normalization layers to ensure stable training.

D. Flow Matching Action Generation

1) Probabilistic Path and Vector Field: The core objective of our action generation module is to establish a deterministic mapping that transforms a simple noise distribution p_0 into a complex robot action distribution p_1. Following the flow matching (FM) paradigm , we define a probability path p_t and a corresponding ground truth vector field $u(t,x)$ that generates this path. The state transition, denoted as a flow $\xi_x(t)$, is governed by the following Ordinary Differential Equation (ODE):

$$\frac{d\xi_x(t)}{dt} = \nu_\theta(t, \xi_x(t)), \quad \xi_x(0) = x, \tag{12}$$

where x represents the initial noise sampled from p_0. Our goal is to train a neural network ν_θ to approximate this vector field by minimizing the flow matching loss:

$$L_{FM}(\theta) = \mathbb{E}_{t,p_t} ||\nu_\theta(t, x_t) - u(t, x_t)||_2^2. \tag{13}$$

2) Action Generation via Consistency Flow Matching: To achieve real-time inference, we leverage the Consistency Flow Matching. Unlike standard FM which requires multi-step ODE solvers, Consistency-FM refines the flow dynamics by enforcing self-consistency in the velocity field, enabling one-step inference. Specifically, we learn a velocity-consistent vector field that defines straight-line flows from any time t to the same endpoint in action space.

For our robot manipulation task, the conditional trajectory generation is formulated as follows:

$$a_{future} = f_\theta(t, a_t, s, v) = a_t + (1-t) \cdot \nu_\theta(t, a_t, s, v), \tag{14}$$

where a_t is the intermediate action trajectory at time t, and the generation is conditioned on the robot state s and visual representation v.

3) Training Objective and Multi-segment Refinement: To ensure high-quality action synthesis while maintaining efficiency, we employ a multi-segment training strategy. By splitting the time horizon into K segments, the model learns segmented linear trajectories that are more flexible than a single straight line. Additionally, we sample an inpainting mask $M \in \{0,1\}^{H \times D_a}$ that reveals partial future action entries, and formulate the inpainted noised sample $\tilde{a}_t$. The final training objective combines a consistency loss and a velocity regularization term:

$$\begin{aligned} L(\theta^*) = &\mathbb{E}_{t,a_t}\Big[\lambda^i \| f_\theta^i(t, \tilde{a}_t, s, v) - f_{\theta^-}^i(t + \Delta t, \tilde{a}_{t+\Delta t}, s, v)\|_2^2 \\ &+ \alpha \| \nu_\theta^i(t, \tilde{a}_t, s, v) - \nu_{\theta^-}^i(t + \Delta t, \tilde{a}_{t+\Delta t}, s, v)\|_2^2 \\ &+ \mathrm{reg}(\theta)\Big], \end{aligned} \tag{15}$$

where θ^- denotes the exponential moving average (EMA) of parameters , and α is a weighting scalar. This ensures that the flow model can decode high-quality robot actions in a single step during inference, effectively breaking the performance bottleneck of recursive diffusion-based methods.

V. EXPERIMENTS

A. Benchmarks and Task Configuration

We evaluate our STAMP's performance on two robotic simulators: Adroit [41] and MetaWorld [42]. Adroit includes three tasks—Pen, Door, and Hammer—focusing on high-dimensional dexterous manipulation with a multi-fingered hand. Meanwhile, MetaWorld consists of 34 tasks for a parallel-jaw robotic arm, assessing policy versatility and generalization. We categorize MetaWorld tasks into four difficulty levels: Easy, Medium, Hard, and Very Hard.

B. Baselines and Evaluation Metrics

We evaluate the trade-off between inference efficiency and task success by comparing with three representative imitation learning baselines:

- **2D diffusion:** Diffusion Policy (DP) [6] as a widely used image-based baseline.
- **3D diffusion:** DP3 [5], a 3D vision-based conditional diffusion policy.
- **Flow-based:** FlowPolicy [28], which leverages consistency flow matching for efficient action generation.

We report **Success Rate** (mean over 20 episodes), **Inference Time** (ms per action sequence), and **NFE** (number of network evaluations per action).

C. Implementation Details

We use an MLP to encode both visual features and proprioceptive features. For 3D inputs, point clouds are downsampled to 1024 points using farthest point sampling (FPS). Temporal context is provided by a history window of k_o=8 for observations and k_a=4 for actions. The policy predicts a 4-step action sequence at each step. Our model uses a conditional 1D U-Net backbone with hierarchical downsampling for flow

TABLE I: Comparisons on success rate (%) between state-of-the-art policy models

Methods	NFE	Adroit			Metaworld				**Average**
		Ham.	Door	Pen	Easy(21)	Medium(4)	Hard(4)	Very Hard(5)	
DP	10	16 ± 10	34 ± 11	13 ± 2	50.7 ± 6.1	11.0 ± 2.5	5.25 ± 2.5	22.0 ± 5.0	35.2 ± 5.3
DP3	10	100 ± 0	56 ± 5	46 ± 10	87.3 ± 2.2	44.5 ± 8.7	32.7 ± 7.7	39.4 ± 9.0	68.7 ± 4.7
FlowPolicy	1	100 ± 0	58 ± 5	53 ± 12	90.2 ± 2.8	$\mathbf{47.5 \pm 7.7}$	37.2 ± 7.2	$\mathbf{36.6 \pm 7.6}$	70.0 ± 4.7
Ours	**1**	$\mathbf{100 \pm 0}$	$\mathbf{62 \pm 7}$	$\mathbf{53 \pm 10}$	$\mathbf{92.3 \pm 2.6}$	46.8 ± 6.5	$\mathbf{38.7 \pm 7.6}$	36.5 ± 9.1	$\mathbf{72.4 \pm 4.7}$

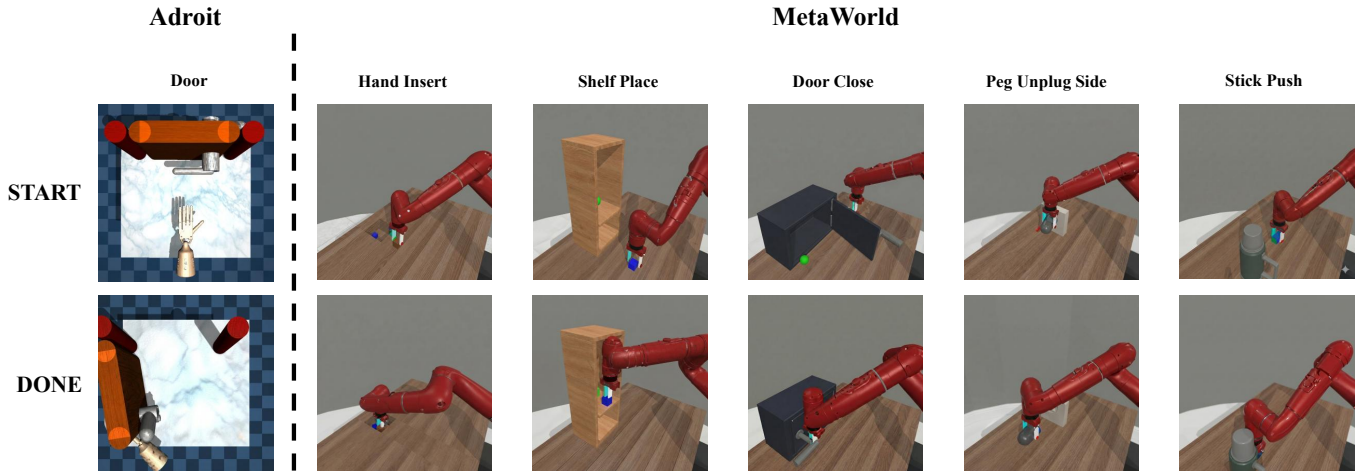

Fig. 4: Visualization of inference. We select several tasks in Adroit and Metaworld.

matching, incorporating FiLM [43] layers for effective integration of temporal memory as conditioning signals. The model is developed using the PyTorch framework, with all experiments (training and inference) conducted on a single NVIDIA RTX A4000 GPU.

D. Comparison Results

1) Task Performance and Inference Efficiency: As shown in Table I, our STAMP outperforms existing 2D and 3D baselines across all 37 tasks, achieving an average success rate of 72.4%, 2.4% higher than the best baseline, FlowPolicy. The unified multimodal framework excels at capturing complementary cues from proprioception and perception, especially for tasks requiring temporal accumulation and stage transitions.

We attribute the improvements to two factors: Pyramid Memory Module, which aggregates multi-granularity temporal context, and Adaptive Fusion Mechanism, which stabilizes conditional generation by gating observations with multi-scale memory and action priors.

As shown in Table II, our STAMP also achieves efficient real-time performance, with a 26.5ms inference time, outperforming DP and DP3. Compared to FlowPolicy (19.9ms), STAMP delivers higher success rates while maintaining a comparable runtime.

2) Qualitative Comparison on Manipulation Tasks: To evaluate the efficacy of STAMP, we conducted extensive inference tests across a diverse set of manipulation tasks. As illustrated in Fig. 4, STAMP consistently accomplishes multiple complex tasks while ensuring safe trajectories in complex environments. The results demonstrate that STAMP can effectively generalize to various task dynamics and maintain robust performance.

TABLE II: Comparisons on inference time per step (ms) between state-of-the-art policy models

Methods	**Adroit**	**Metaworld**	**Average**
DP	100.3 ± 2.6	106.5 ± 2.0	105.9 ± 2.1
DP3	146.1 ± 1.9	145.7 ± 2.3	145.7 ± 2.3
FlowPolicy	20.1 ± 0.5	19.9 ± 0.1	$\mathbf{19.9 \pm 0.2}$
Ours	26.8 ± 0.7	26.5 ± 0.4	26.5 ± 0.4

Learning curves for four representative tasks are illustrated in Fig. 5, where STAMP consistently surpasses baselines in overall success rate. Notably, STAMP achieves the fastest convergence in the 'Assembly' task. On medium-difficulty tasks such as 'Reach Wall' and 'Door', our model exhibits steady improvement, whereas FlowPolicy and DP3 tend to fluctuate or plateau. Even in the more challenging 'Soccer' task, STAMP maintains superior performance and stability, demonstrating its robustness in handling complex manipula-

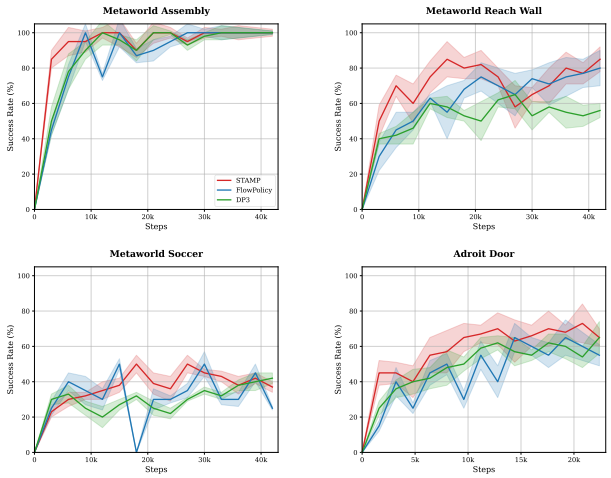

Fig. 5: Illustrations of the learning curves.

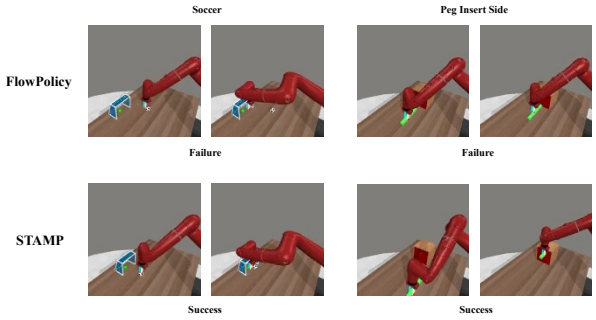

Fig. 6: Comparison cross tasks.

tion dynamics.

Fig. 6 illustrates the qualitative performance of STAMP and FlowPolicy in the 'Soccer' and 'Peg Insert Side' tasks. The 'Soccer' task involves precise interactions with a dynamic small object. While FlowPolicy often fails due to temporal inconsistencies, the intrinsic memory mechanism in STAMP ensures consistent state backtracking and updating, leading to successful execution. In the 'Peg Insert Side' scenario, where the target hole is partially occluded, FlowPolicy struggles with imprecise insertion. In contrast, STAMP leverages its adaptive gating mechanism to effectively complement current observations with historical memory, thereby maintaining high precision under limited visibility.

E. Ablation Studies

1) Ablation study on components: To assess the contribution of each individual component within STAMP, we conducted a systematic ablation study, as summarized in Table III. Starting from a baseline generative policy, the integration

TABLE III: Ablation study of proposed components of STAMP

Memory Pyramid	Adaptive Gating	Action History	Adroit	MetaWorld	**Average**
			65.8	67.5	67.4
✓			68.6	70.0	69.9
✓	✓		70.4	71.8	71.7
✓	✓	✓	71.6	72.5	**72.4**

TABLE IV: Impact of history length

History Length (L)	**Adroit**	**MetaWorld**	**Average**
$L = 3$	69.2	70.8	70.6
$L = 5$	70.7	71.9	71.8
$L = 7$	71.6	72.5	**72.4**

of the Memory Pyramid leads to a significant performance boost (+2.5% on average), highlighting the importance of temporal context in resolving memory collapse. The addition of the Adaptive Gating mechanism further refines the fusion of features, yielding an average success rate 71.7%. Finally, incorporating Action History completes our complete architecture, achieving the maximum performance of 72.4%. These results empirically validate that each proposed module plays a complementary role in enhancing the policy's robustness for complex manipulation tasks.

2) Analysis of Temporal Window Size: To investigate the impact of temporal context, we conducted an ablation study on the size of the history window L. As illustrated in Table IV, the performance across both benchmarks scales positively with the length of the history observation. Specifically, increasing L from 3 to 7 produces a steady improvement in the average success rate (from 71.6% to 72.4%), validating the necessity of long-horizon temporal information for resolving ambiguities in contacts-rich tasks.

VI. CONCLUSION

In this work, we have introduced a novel paradigm for robotic imitation learning by incorporating insights from the hierarchical nature of human memory. We have developed STAMP, a robotic manipulation policy based on flow matching. The core of STAMP lies in its cognitive-inspired memory pyramid, which enables multi-granularity feature aggregation and prevents memory collapse in long-horizon tasks. This 4D spatiotemporal framework allows the robot to maintain a coherent understanding of world states, effectively aligning perceptual geometry with action generation. Our evaluations on the Adroit and MetaWorld benchmarks demonstrate that STAMP achieves a favorable trade-off between modeling complexity and runtime overhead.

REFERENCES

[1] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[2] T. Feng, X. Wang, Y.-G. Jiang, and W. Zhu, "Embodied ai: From llms to world models," *arXiv preprint arXiv:2509.20021*, 2025.

[3] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi *et al.*, "Openvla: An open-source vision-language-action model, 2024," *URL https://arxiv.org/abs/2406.09246*, 2024.

[4] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, "Implicit behavioral cloning," in *Conference on robot learning*. PMLR, 2022, pp. 158–168.

[5] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," *arXiv preprint arXiv:2403.03954*, 2024.

[6] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, vol. 44, no. 10-11, pp. 1684–1704, 2025.

[7] H. Shi, B. Xie, Y. Liu, L. Sun, F. Liu, T. Wang, E. Zhou, H. Fan, X. Zhang, and G. Huang, "Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation," *arXiv preprint arXiv:2508.19236*, 2025.

[8] C. Wang, X. Luo, K. Ross, and D. Li, "Vrl3: A data-driven framework for visual deep reinforcement learning," in *Conference on Neural Information Processing Systems*, 2022. [Online]. Available: https://openreview.net/forum?id=NjKAm5wMbo2

[9] C. Wang, H. Fang, H.-S. Fang, and C. Lu, "Rise: 3d perception makes real-world robot imitation simple and effective," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 2870–2877.

[10] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[11] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3929–3945, 2023.

[12] S. Cheng and D. Xu, "League: Guided skill learning and abstraction for long-horizon manipulation," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6451–6458, 2023.

[13] M. Lei, H. Cai, Z. Cui, L. Tan, J. Hong, G. Hu, S. Zhu, Y. Wu, S. Jiang, G. Wang *et al.*, "Robomemory: A brain-inspired multi-memory agentic framework for lifelong learning in physical embodied systems," in *NeurIPS 2025 Workshop on Space in Vision, Language, and Embodied AI*, 2025.

[14] Y. Guo, L. X. Shi, J. Chen, and C. Finn, "Ctrl-world: A controllable generative world model for robot manipulation," *arXiv preprint arXiv:2510.10125*, 2025.

[15] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "π0: A vision-language-action flow model for general robot control. corr, abs/2410.24164, 2024. doi: 10.48550," *arXiv preprint ARXIV.2410.24164*.

[16] J. Wen, Y. Zhu, J. Li, Z. Tang, C. Shen, and F. Feng, "Dexvla: Vision-language model with plug-in diffusion expert for general robot control," *arXiv preprint arXiv:2502.05855*, 2025.

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

[18] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[19] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," *arXiv preprint arXiv:2210.02747*, 2022.

[20] L. Yang, Z. Zhang, Z. Zhang, X. Liu, M. Xu, W. Zhang, C. Meng, S. Ermon, and B. Cui, "Consistency flow matching: Defining straight flows with velocity consistency," *arXiv preprint arXiv:2407.02398*, 2024.

[21] A. Prasad, K. Lin, J. Wu, L. Zhou, and J. Bohg, "Consistency policy: Accelerated visuomotor policies via consistency distillation," *arXiv preprint arXiv:2405.07503*, 2024.

[22] Q. Bu, Y. Yang, J. Cai, S. Gao, G. Ren, M. Yao, P. Luo, and H. Li, "Univla: Learning to act anywhere with task-centric latent actions," *arXiv preprint arXiv:2505.06111*, 2025.

[23] W. Song, Z. Zhou, H. Zhao, J. Chen, P. Ding, H. Yan, Y. Huang, F. Tang, D. Wang, and H. Li, "Reconvla: Reconstructive vision-language-action model as effective robot perceiver," *arXiv preprint arXiv:2508.10333*, 2025.

[24] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.

[25] L. Sun, B. Xie, Y. Liu, H. Shi, T. Wang, and J. Cao, "Geovla: Empowering 3d representations in vision-language-action models," *arXiv preprint arXiv:2508.09071*, 2025.

[26] D. Qu, H. Song, Q. Chen, Y. Yao, X. Ye, Y. Ding, Z. Wang, J. Gu, B. Zhao, D. Wang *et al.*, "Spatialvla: Exploring spatial representations for visual-language-action model," *arXiv preprint arXiv:2501.15830*, 2025.

[27] X. Hu, Q. Liu, X. Liu, and B. Liu, "Adaflow: Imitation learning with variance-adaptive flow-based policies," *Advances in Neural Information Processing Systems*, vol. 37, pp. 138 836–138 858, 2024.

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

[29] W.-D. Chang, J. C. G. Higuera, S. Fujimoto, D. Meger, and G. Dudek, "Il-flow: Imitation learning from observation using normalizing flows," *arXiv preprint arXiv:2205.09251*, 2022.

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

[31] J. Feng, J. Lee, S. Geisler, S. Gunnemann, and R. Triebel, "Topology-matching normalizing flows for out-of-distribution detection in robot learning," *arXiv preprint arXiv:2311.06481*, 2023.

[32] G. Lu, Z. Gao, T. Chen, W. Dai, Z. Wang, W. Ding, and Y. Tang, "Manicm: Real-time 3d diffusion policy via consistency model for robotic manipulation," *arXiv preprint arXiv:2406.01586*, 2024.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] X. Li, M. Liu, H. Zhang, C. Yu, J. Xu, H. Wu, C. Cheang, Y. Jing, W. Zhang, H. Liu *et al.*, "Vision-language foundation models as effective robot imitators," *arXiv preprint arXiv:2311.01378*, 2023.

[35] Q. Zhao, Y. Lu, M. J. Kim, Z. Fu, Z. Zhang, Y. Wu, Z. Li, Q. Ma, S. Han, C. Finn *et al.*, "Cot-vla: Visual chain-of-thought reasoning for vision-language-action models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 1702–1713.

[36] G. Lu, B. Jia, P. Li, Y. Chen, Z. Wang, Y. Tang, and S. Huang, "Gwm: Towards scalable gaussian world models for robotic manipulation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 9263–9274.

[37] J. Cen, C. Yu, H. Yuan, Y. Jiang, S. Huang, J. Guo, X. Li, Y. Song, H. Luo, F. Wang *et al.*, "Worldvla: Towards autoregressive action world model," *arXiv preprint arXiv:2506.21539*, 2025.

[38] W. Li, R. Zhang, R. Shao, J. He, and L. Nie, "Cogvla: Cognition-aligned vision-language-action model via instruction-driven routing & sparsification," *arXiv preprint arXiv:2508.21046*, 2025.

[39] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[41] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," *arXiv preprint arXiv:1709.10087*, 2017.

[42] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on robot learning*. PMLR, 2020, pp. 1094–1100.

[43] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.