2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Seeing Motion, Generating Action: Explicit Motion-Aware Policy for Robotic Action Generation

Yixiong Li[1], Ye Zhang[1], Yun Pei[1,2], Yongjian Zhang[1], Ruimao Zhang[1], Yulan Guo[1,†]

***Abstract*— Imitation learning (IL) offers a scalable framework for teaching robots complex manipulation skills from human demonstrations. However, conventional end-to-end visuomotor IL models often suffer from poor performance and robustness due to the significant modality mismatch between high-dimensional visual inputs and low-dimensional motor actions. The redundant information in RGB image, such as color of ambient light, leads models to depend on strong yet brittle task irrelevant priors that ultimately degrade performance across diverse visual environments. To address these limitations, we propose Motion-Aware Two-Stream Policy (MTP) – a novel imitation learning architecture that explicitly incorporates motion priors via optical flow alongside RGB observations. MTP employs a two-stream perception module that separately encodes spatial (RGB) and temporal (optical flow) information. These spatial-temporal features are fused and fed into a conditional flow matching module to generate actions. We evaluate MTP extensively in both simulation and real-world robot tasks. Results show that MTP significantly outperforms state-of-the-art baselines in terms of success rate and robustness to visual perturbations, demonstrating its effectiveness in generalizable robotic manipulation.**

I. INTRODUCTION

Imitation learning (IL) offers a scalable framework for teaching robots complex manipulation skills, such as grasping [1], [2], mobile manipulation [3], [4] and dexterous manipulation [5]. Several prominent methods, such as ACT [6] and Diffusion Policy [7] have achieved significant success in this domain. However, the performance and robustness of end-to-end imitation learning policies are often limited due to the significant modality mismatch between sensory inputs and motor outputs [8] and the redundancy in image representation [9], [10]. Visual inputs are typically high-dimensional, noisy, static, and semantically ambiguous, while robotic actions are low-dimensional, dynamic, and task-specific. Directly mapping from image to robot action (Fig. 1a) will complicate the learning process, as models must bridge the gap between perception and control. A simple way to reduce the gap between perception and control is using multiple consecutive visual observations as input (Fig. 1b), since successive image frames contain motion information about the scene, which is intrinsically aligned with action semantics. However, using multiple consecutive visual observations may, in some cases, degrade performance [11], or even result in training instability or failure. This degradation occurs because the model must implicitly learn to extract motion dynamics from multi-step observations. It is particularly challenging when training data is limited or suboptimal. Moreover, visual observations often contain substantial redundancy and numerous irrelevant features, such as textures and lighting variations, which hinder the model's ability to extract salient, task-relevant information. Such redundancy can also cause the model to overfit to superficial visual cues and overspecify task goals with unnecessary detail [9], [10], thereby introducing strong prior assumptions that ultimately degrade generalization performance across diverse visual environments.

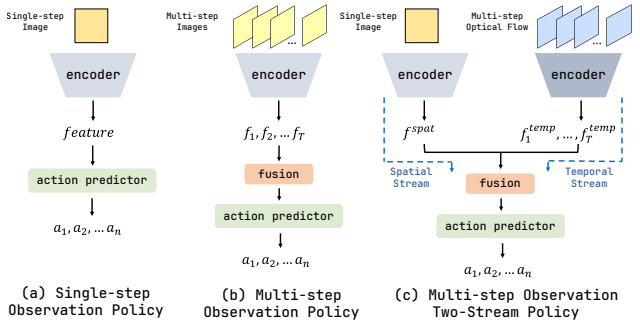

Fig. 1. Three Different Architectures. (a) The model receives single-step image observation and encode it into a feature which is then passed to the action predictor to generate the action. (b) The model receives multi-step image observations and encoder each image into a feature representation. These features are then fused by a fusion module and the resulting fused feature is used by action predictor to predict the action. (c) The spatial stream processes a single-step image observation and encodes it into a feature while the temporal stream processes multi-step optical flow extracted from past image observations and encodes them into temporal features. The features from both streams are fused via a fusion module and the fused feature is used by action predictor for final action prediction.

To address these limitations, we propose **MTP** (**M**otion-Aware **T**wo-Stream **P**olicy), a two-stream architecture (Fig. 1c) imitation learning algorithm which is augmented with explicit motion priors derived from optical flow. The spatial stream maps raw image observations into a latent embedding, which encodes the spatial information of the scene. The temporal stream handles the multi-step optical flow extracted from raw image observation and extracts features respectively using a visual encoder. To prevent the spatial features from being dominated by the multi-step temporal features, a custom **TempoFormer** module is designed to fuse the temporal features. Finally, the MTP employs conditional flow-matching (CFM) conditioned on fused spatial-temporal feature to predict the robot action. The two-stream architec-

†Yulan Guo is the corresponding author.
[1]School of Electronics and Communication Engineering, the Shenzhen Campus of Sun Yat-sen University, China.
[2]Pengcheng Laboratory, Shenzhen, China.
{zhangy2658,zhangrm27,guoyulan}@mail.sysu.edu.cn.
{liyx357,peiy8,zhangyj85}@mail2.sysu.edu.cn.

ture offers several advantages:

- **Reduce the modality mismatch between visual inputs and motor outputs**. It provides a compact representation of scene dynamics that is intrinsically aligned with action semantics.
- **Directly process scene dynamics from past observations**. It eliminates the need to implicitly learn motion representations from raw image sequences and thereby improving training efficiency.
- **Enhances the model's robustness to visual changes**. The appearance invariance of optical flow enhances the model's robustness to visual changes in the environment, such as variations in lighting.

In summary, the contributions of our paper are summarized as follows:

1) We propose a novel two-stream architecture that incorporates spatial and temporal streams, which achieves better spatial-temporal relationship understanding.
2) We develop an efficient **TempoFormer** module, which can effectively interact and fuse information of temporal features.
3) Extensive simulations and real-world experiment results demonstrate that our method achieves the state-of-art performance.

II. RELATED WORKS

A. Imitation Learning

In recent years, Behavior Cloning has resurged with the advent of deep learning, particularly in robotic manipulation tasks like grasping and assembly [7], [12], [13], [14], [15], [16], [6], [17], [18]. Large datasets [19], [20], [21] of expert trajectories now train deep neural networks to capture intricate motion patterns. For example, ACT [6] employs a Conditional Variational Autoencoder [22]) combined with Transformer architectures [23] and ResNet image encoders [24] to model the variability of human demonstrations, enabling temporally extended action sequences. Meanwhile, Diffusion Policy [7] leverages diffusion processes [25], [26] to generate multimodal action distributions, effectively handling ambiguous or diverse human strategies. Recently, conditional flow-matching (or rectified flow) has been proposed as a more flexible generalization of diffusion models. Several works [27], [14] utilize flow-matching model [28], [29], [30] as the action predictor to improve the model's ability to fit distributions.

B. Motion Representations for Visuomotor Policies

To incorporate motion information into visuomotor policies, recent works exploit the idea of using flow for robot action prediction. For instance, FlowBotHD [31] predicts 3D articulation flow from point clouds to manipulate articulated objects, while General Flow [32] learns to predict future 3D point trajectories from human RGB-D videos, treating it as a scalable affordance. Img2Flow2Act [33] uses predicted object trajectories as a cross-domain task interface, and FabricFlowNet [34] uses optical flow to represent the correspondence between the current and target cloth configurations. In contrast to the above methods which predict abstract future flows or use them as a task objective, our work revisit classic, dense 2D optical flow and treat it as a direct physical observation. Experiments suggest that this directly observed physical motion signal can serve as a powerful, model-free prior to enhance the policy's immediate dynamic understanding and its robustness to visual changes, all while requiring only a standard RGB camera.

III. METHOD

A. Problem Definition

We assume a dataset $\mathcal{D} = \{D_1, D_2, \cdots, D_n\}$ of n expert demonstrations covering various tasks is given. Each demonstration $D_i = (\{o^i_{1\ldots m_i}\}, \{a^i_{1\ldots m_i}\})$ is a successful roll-out of length m_i, $\{o^i_1, o^i_2, \ldots, o^i_{m_i}\}$ is a sequence of the observations and $\{a^i_1, a^i_2, \ldots, a^i_{m_i}\}$ is the sequence of corresponding actions. Each observation $o_t = \{I_t, P_t\}$ contains images I_t and robot proprioception P_t. We want to learn a visuomotor policy $\pi : \mathcal{O} \rightarrow \mathcal{A}$ that maps the visual observations $o \in \mathcal{O}$ to actions $a \in \mathcal{A}$, such that our robots not only reproduce the skill but also generalize beyond the training data.

To this end, we introduce Motion-Aware Two-Stream Policy (MTP), which mainly consists of three critical parts: (a) **Spatial Stream**. MTP perceives the environments with images and these visual observations are fed into visual encoder to extract visual features. Then these visual observation are stored which will be used in the temporal stream. (b) **Temporal Stream**. Firstly, optical flow are extracted from the past observations using opencv TV-L1 optical flow [35]. Then each optical flow image is fed into a weight shared visual encoder to extract a feature. Finally, a **TempoFormer** module is designed for effectively fusing the information between multi-step optical flow features. (c) **Action Prediction**. MTP utilizes the Flow Matching Policy [14] as the action-making backbone, which generates action sequences conditioning on our features. An overview of MTP is in Fig. 2. We will detail each module in the following sections.

B. Spatial Stream

The spatial stream maps the raw image observations into a latent embedding $f^{spat} \in \mathbb{R}^{d_s}$ and the raw image observations are stored for utilization in temporal stream processing. The latent feature f^{spat} encode the spatial information of the scene. Throughout the end-to-end training process, the neural network establishes a mapping from the scene to robot action, which means that the network can infer the robot action from the given scene.

We use a standard ResNet18 [24] as the visual encoder with the modifications as: (1) Replace the global average pooling with a spatial softmax pooling. (2) Replace BatchNorm with GroupNorm [36] which is important for stable training [7].

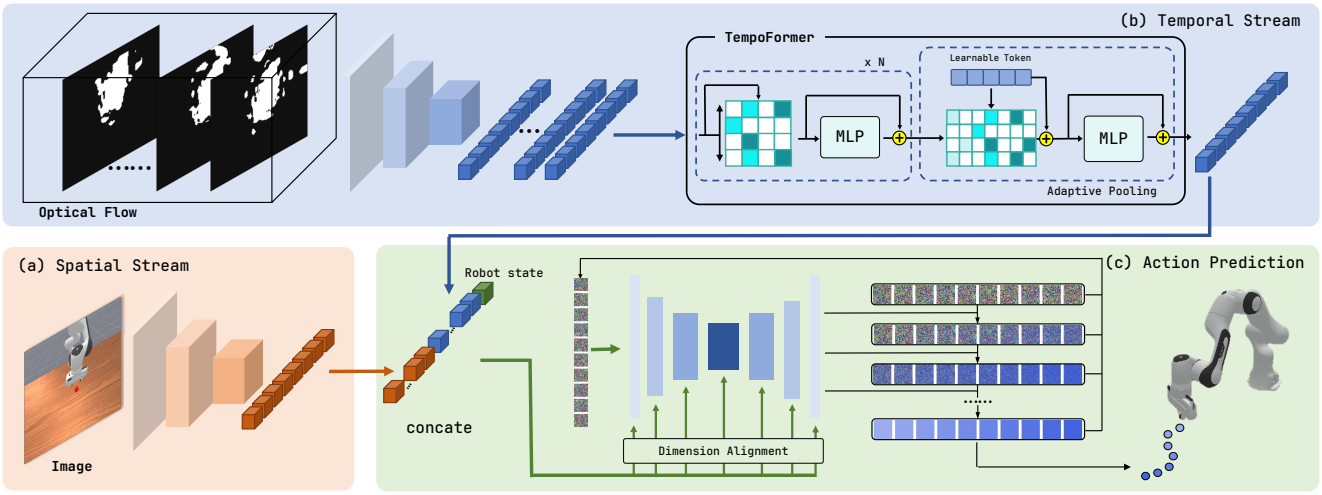

Fig. 2. Overview of MTP. (a) **Spatial Stream**. MTP perceives the environments with images and these visual observations are fed into visual encoder into visual features. (b) **Temporal Stream**. Firstly, optical flow are extracted from the past observations using opencv TV-L1 optical flow[35]. Then each optical flow image is fed into a visual encoder to extract a feature while all optical flow images share the same visual encoder and weight. Finally, a **TempoFormer** module is designed for effectively fuse the information between multi-step optical flow features. (c) **Action Prediction**. MTP utilizes the expressive Flow Matching Policy as the action-making backbone to generates action sequences conditioning on our spatial and temporal features.

C. Temporal Stream

The temporal stream encode the history information to enhance the network's performance. However, image contains too much redundant information which make it hard to extract the effective history information. The most valuable and the most information we want to use is the change of the scene. For this insight, we use optical flow as the representation which naturally contains the motion information. By computing per-pixel displacement vector fields between consecutive frames, it enables several key advantages. First, optical flow provides a physics-grounded and compact representation of scene dynamics that is intrinsically aligned with action semantics. Second, optical flow **explicitly encodes motion information** by quantifying how agents or objects move relative to the observer, thereby directly correlating with executed actions. Third, it **suppresses static distractors**, as stationary pixels exhibit near-zero flow, naturally filtering out textures, shadows, and background clutter that typically hinder RGB-based policies. Finally, optical flow **exhibits strong appearance invariance**, being robust to variations in color, texture, and illumination, which enhances generalization to visual domain shifts, such as changes in lighting conditions.

In the temporal stream, firstly, we load $T+1$ steps image observations from the past observations and extract T steps optical flow frames $F_1, F_2, \ldots, F_T \in \mathbb{R}^{H\times W}$ between consecutive image observations using OpenCV TV-L1 optical flow algorithm [35]. We use a standard ResNet18 [24] as the optical flow encoder with the same modifications described in III-B. All optical flow frames across the observation sequence share the same encoder with tied weights and a feature is extracted from each optical flow.

$$[f_1^{opt}, f_2^{opt}, \ldots, f_T^{opt}] = Encoder([F_1, F_2, \ldots, F_T]) \quad (1)$$

To prevent the spatial features from being dominated by the multi-step temporal features, we design a **TempoFormer** module which exchanges the information between the different step of optical flow features and and output the temporal aggregated features. Firstly, T steps optical flow features $\{f_1^{opt}, f_2^{opt}, \ldots, f_T^{opt}\}$ are stack as a sequence of length T and the feature sequence is processed through N feature interaction blocks to enable deeper fusion. Each block first applies multi-head self-attention to exchange information dimension-by-dimension followed by a MLP for feature transformation. After that, we obtain a feature sequence $L \in \mathbb{R}^{T\times d_o}$ with sufficient information interaction.

We design an adaptive pooling module to compress feature sequence into a $\mathbb{R}^{d_o}$ feature vector. The module receives the feature sequence L output by the feature interaction block as keys and values of multi-head cross-attention block. A learnable token $[\mathrm{CLS}] \in \mathbb{R}^{d_o}$ will be used as a query for multi-head cross-attention to control the output dimension. The MLP block outputs a feature vector f^{opt} containing multi-step optical flow information.

D. Action Prediction

Conditional Flow Matching. The decision module in MTP employs a conditional flow-matching (CFM) framework [29], [28] conditioned on fused features. This approach formulates action prediction through a continuous-time ordinary differential equation:

$$\frac{d}{dt}\mathbf{z}(t) = \mathbf{v}(\mathbf{z}(t), t), \quad \mathbf{z}(t=0) \sim p_0 \quad (2)$$

where p_0 denotes the initial noise distribution (typically isotropic Gaussian $\mathcal{N}(0,1)$), and $\mathbf{v} : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ defines a time-dependent velocity field that evolves samples from p_0 to the target data distribution. The learning objective focuses on discovering the optimal vector field through minimization:

$$\min_{\mathbf{v}} \mathbb{E}_{\mathbf{z}_0\sim p_0, \mathbf{z}_1\sim p_1}\Big[\int_0^1 \|\mathbf{v}_{gt}(\mathbf{z}_1, \mathbf{z}_0) - \mathbf{v}_\theta(\mathbf{z}_t, t)\|_2^2 dt\Big] \quad (3)$$

where $\mathbf{v_{gt}}$ is the choice of ground truth vector field we regress against. CFM's key insight lies in its regression target – a straight-line vector field $\mathbf{v}_{gt} = \mathbf{z}_1 - \mathbf{z}_0$ that generates linear probability paths between noise and data samples. For policy learning, the target samples $\mathbf{z_1}$ correspond to trajectories from expert demonstrations. The velocity field $\mathbf{v}_\theta$ is implemented as a neural network trained via gradient descent on Eq. 3.

During inference, trajectory generation follows a discrete integration scheme:

$$\mathbf{z}_t = \mathbf{z}_{t-1} + \mathbf{v}_\theta(\mathbf{z}_{t-1}, t)dt, \quad \mathbf{z}_0 \sim \mathcal{N}(0,1), dt = 1/N_{step} \tag{4}$$

where N_{step} controls the numerical integration precision. This formulation enables efficient sampling of robot trajectories that match the expert behavior distribution.

Implementation details. Similar to [7], the model used to de-noise trajectories is a conditional 1D U-Net and it employs FiLM [37] as conditioning mechanics for the observation inputs. The model takes random noise samples as input and conditions on the concatenated features from the temporal and spatial streams. In the case of CFM, it predicts the velocity by sampling from the learned distribution conditioned on this feature.

IV. SIMULATED EXPERIMENTS

A. Setup

Platform. We evaluate our proposed MTP on Maniskill [38], a powerful unified framework for robot simulation. Each environment consists of the 7-DoF Franka Emika robot arm to execute actions and a single-view camera in front of the workspace. We use the dataset released in Maniskill's benchmarks which are collected by motion planning (PushCube-v1) or PPO [39] (PokeCube-v1, PullCube-v1, LifePegUpright-v1, RollBall-v1) and each task includes non-overlapping 50 episodes for training.

Baselines. Taking into account the task paradigm and data modality, we choose the ACT [6], DP (Diffusion Policy [7]) and FP (Flow Matching Policy [14]) as baselines. Diffusion Policy and ACT is directly adopted from the source repository without any modifying of hyperparameters. Flow Matching Policy is a point cloud based policy and we use the image edition provided in its source repository without any modifying of hyperparameters.

Evaluation metric. We train the policies for each experiment with seed number 42. While performing evaluation, we set the seeds in the environment starting from 10000 which is different from that in dataset and randomly initialize the environment state for generating a environment state that different from that in dataset, aiming at evaluating the generalizability of the methods. Since the baselines and our method are generative-model-based policy, we evaluate the methods using three random seeds (22, 42, 62) to ensure diversity in initialization. We record the success rate and report mean and standard deviation.

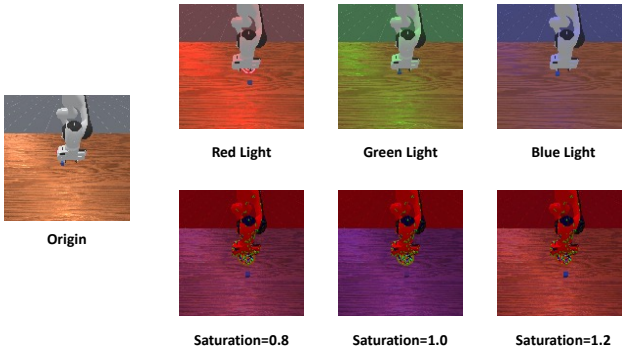

Fig. 3. Visual Variation Example. The three figures above demonstration the effect of changing the background lighting color from white to red / green / blue. The rest at the bottom give some examples for saturation jitter. Only 3 examples are given here, and 10 saturation jitter factors are actually used in the evaluation.

B. Benchmarking Result

To evaluate the robustness and generalization capabilities of our method, we conduct experiments under randomized initialization conditions. Each task is evaluated over 50 independent trials. To ensure statistical reliability and mitigate the effects of randomness, for each of the five tasks, we conduct 10 evaluation runs across three random seeds, resulting in a total of 150 experiments per method, and then report the mean and standard deviation of the success rate.

As shown in Tab. I, our method achieves the highest average success rate (63.1%) across all five benchmark tasks, outperforming existing baselines. Notably, our approach dominates in three manipulation-intensive tasks (PokeCube-v1, PullCube-v1, RollBall-v1), where precise object interaction is critical. In the remaining two tasks (LiftPegUpright-v1, PushCube-v1), our method attains competitive second-place performance. This consistent improvement in diverse scenarios highlights our method's enhanced generalization capabilities for policy learning.

C. Visual Variation Evaluation

To evaluate the visual robustness of the policy, we conduct the two types of experiments: (1) Change the background directional light from white to red / green / blue to simulate ambient lighting changes. (2) Change the saturation of the image observation to simulate extreme visual situation. The saturation factor of the image observation varies from 0.8 to 1.25 with the step of 0.05, totally 10 values. The visual variation effect is demonstrated in Fig. 3. The results of the experiments are shown at Tab. II.

As the result shown in Tab. II, MTP achieves the best performance at green ambient light and blue ambient light evaluation and significantly outperforms the second best method (MTP achieves a 36.6% relative improvement in average success rate under green ambient light conditions and 58.9% under blue ambient light evaluation). MTP achieves second best method performance at red ambient light, but our average success rate is only 1.5% lower than the highest success rate. Compared with other methods whose success

TABLE I

BENCHMARKING RESULT

Method \ Task	LiftPegUpright-v1	PokeCube-v1	PullCube-v1	PushCube-v1	RollBall-v1	Mean SR (↑)
ACT [6]	$0.14^{\pm 0.003}$	$0.346^{\pm 0.0006}$	$0.366^{\pm 0.003}$	$\mathbf{0.953^{\pm 0.0006}}$	$0.113^{\pm 0.0}$	0.384
DP [7]	$0.645^{\pm 0.036}$	$0.643^{\pm 0.042}$	$\underline{0.783}^{\pm 0.018}$	$0.751^{\pm 0.026}$	$\underline{0.143}^{\pm 0.023}$	$\underline{0.593}$
FP [14]	$\mathbf{0.729^{\pm 0.048}}$	$\underline{0.645}^{\pm 0.036}$	$0.771^{\pm 0.027}$	$0.658^{\pm 0.037}$	$0.119^{\pm 0.028}$	0.582
MTP (ours)	$\underline{0.652}^{\pm 0.04}$	$\mathbf{0.655^{\pm 0.034}}$	$\mathbf{0.837^{\pm 0.027}}$	$\underline{0.838}^{\pm 0.031}$	$\mathbf{0.172^{\pm 0.036}}$	**0.631**

TABLE II

VISUAL VARIATION EVALUATION

Method \ Task		LiftPegUpright-v1	PokeCube-v1	PullCube-v1	PushCube-v1	RollBall-v1	Mean SR(↑)
Red Light	ACT [6]	$0.127^{\pm 0.025}$	$0.247^{\pm 0.030}$	$0.413^{\pm 0.045}$	$0.021^{\pm 0.029}$	$0.076^{\pm 0.009}$	0.177
	DP [7]	$\underline{0.286}^{\pm 0.029}$	$0.517^{\pm 0.017}$	$0.690^{\pm 0.021}$	$0.127^{\pm 0.030}$	$\underline{0.153}^{\pm 0.020}$	0.355
	FP [14]	$\mathbf{0.654^{\pm 0.037}}$	$\mathbf{0.538^{\pm 0.037}}$	$\underline{0.753}^{\pm 0.031}$	$\underline{0.298}^{\pm 0.040}$	$0.127^{\pm 0.023}$	**0.474**
	MTP (ours)	$0.178^{\pm 0.026}$	$\underline{0.537}^{\pm 0.045}$	$\mathbf{0.769^{\pm 0.027}}$	$\mathbf{0.647^{\pm 0.031}}$	$\mathbf{0.164^{\pm 0.036}}$	$\underline{0.459}$
Green Light	ACT [6]	$0.027^{\pm 0.013}$	$0.185^{\pm 0.038}$	$0.309^{\pm 0.028}$	$0.155^{\pm 0.113}$	$0.061^{\pm 0.023}$	0.147
	DP [7]	$0.071^{\pm 0.024}$	$0.552^{\pm 0.024}$	$0.267^{\pm 0.040}$	$0.145^{\pm 0.025}$	$0.110^{\pm 0.030}$	0.229
	FP [14]	$\underline{0.115}^{\pm 0.024}$	$\mathbf{0.635^{\pm 0.038}}$	$\underline{0.317}^{\pm 0.029}$	$\underline{0.347}^{\pm 0.034}$	$\underline{0.119}^{\pm 0.022}$	$\underline{0.306}$
	MTP (ours)	$\mathbf{0.138^{\pm 0.030}}$	$\underline{0.582}^{\pm 0.043}$	$\mathbf{0.423^{\pm 0.026}}$	$\mathbf{0.765^{\pm 0.043}}$	$\mathbf{0.179^{\pm 0.022}}$	**0.418**
Blue Light	ACT [6]	$0.104^{\pm 0.018}$	$0.195^{\pm 0.046}$	$0.332^{\pm 0.056}$	$\underline{0.463}^{\pm 0.293}$	$0.031^{\pm 0.010}$	0.225
	DP [7]	$0.093^{\pm 0.018}$	$0.406^{\pm 0.047}$	$0.318^{\pm 0.034}$	$0.138^{\pm 0.025}$	$\underline{0.108}^{\pm 0.022}$	0.213
	FP [14]	$\underline{0.154}^{\pm 0.026}$	$\underline{0.522}^{\pm 0.070}$	$\underline{0.357}^{\pm 0.051}$	$0.331^{\pm 0.031}$	$0.095^{\pm 0.020}$	$\underline{0.292}$
	MTP (ours)	$\mathbf{0.205^{\pm 0.035}}$	$\mathbf{0.592^{\pm 0.043}}$	$\mathbf{0.598^{\pm 0.040}}$	$\mathbf{0.778^{\pm 0.041}}$	$\mathbf{0.147^{\pm 0.025}}$	**0.464**
Saturation Jitter	ACT [6]	0.026	0.167	$\underline{0.349}$	0.0	0.014	0.111
	DP [7]	$\underline{0.054}$	0.421	0.259	0.068	0.038	0.168
	FP [14]	0.052	$\underline{0.462}$	**0.352**	$\underline{0.206}$	$\underline{0.107}$	$\underline{0.236}$
	MTP (ours)	**0.058**	**0.575**	0.248	**0.459**	**0.142**	**0.296**

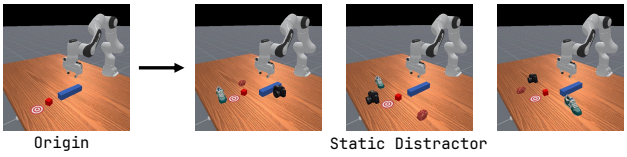

Fig. 4. Static Distractors Examples.

rates fluctuate greatly, the stable and high success rate of MTP shows that our method is very robust to visual variation.

The result of visual saturation jitter evaluation is shown at Tab. II. In such a extreme situation, ACT and Diffusion Policy even fail at PushCube-v1 and RollBall-v1, but MTP still works and achieves a success rate of 45.9% and 14.2% respectively. Our method achieves a success rate of 57.5% and 14.2% at PokeCube-v1 and RollBall-v1 respectively, which is close to the performance at the origin situation.

The visual variation evaluation demonstrate that with the addition of the motion information brought by optical flow, the method can be more robust to visual changes and achieves a better and more stable performance.

D. Static Distractor Evaluation

We also conduct a static distractors experiments to evaluate the robustness of our method. The static distractors effect is demonstrated in Fig. 4. Some irrelevant objects are placed on the workspace as the static distractors. As results shown at Tab. III, our method performs best in these experiments, demonstrating its greater robustness compared to the baseline methods.

E. Ablation Study

1) Ablation Experiments on Architectures: In this experiment, we compare the effectiveness of three architectures, as illustrated in Fig. 1. We modify the Diffusion Policy (DP) [7] and Flow Matching Policy (FP) [14] to adapt them to the different architectures. For the single-step observation policy, we set the observation step in both DP and FP to 1, while the remaining components, such as the encoder, are aligned with our MTP architecture. In the multi-step observation policy, the model receives multi-step observations, which are encoded into a feature sequence using the same encoder. The observation step in DP and FP is aligned with the configuration of MTP. After the visual encoder extracts features from the image sequence, concatenation is used as the fusion method to combine the feature sequence. In the multi-step observation two-stream policy, the spatial stream receives the latest observation, while the temporal stream processes multi-step optical flow derived from past observations. The horizon of the optical flow is aligned with that of MTP. Concatenation

TABLE III

STATIC DISTRACTORS EVALUATION

Method \ Task	LiftPegUpright-v1	PokeCube-v1	PullCube-v1	PushCube-v1	RollBall-v1	Mean SR (↑)
ACT [6]	$0.000^{\pm 0.000}$	$0.009^{\pm 0.008}$	$0.000^{\pm 0.000}$	$0.435^{\pm 0.038}$	$0.000^{\pm 0.000}$	0.089
DP [7]	$0.087^{\pm 0.014}$	$0.565^{\pm 0.051}$	$0.075^{\pm 0.022}$	$\underline{0.669}^{\pm 0.032}$	$\underline{0.129}^{\pm 0.023}$	$\underline{0.305}$
FP [14]	$\underline{0.089}^{\pm 0.028}$	$\underline{0.573}^{\pm 0.039}$	$\underline{0.087}^{\pm 0.019}$	$0.585^{\pm 0.037}$	$0.122^{\pm 0.025}$	0.291
MTP (ours)	$\mathbf{0.178^{\pm 0.025}}$	$\mathbf{0.606^{\pm 0.032}}$	$\mathbf{0.115^{\pm 0.020}}$	$\mathbf{0.793^{\pm 0.029}}$	$\mathbf{0.173^{\pm 0.032}}$	**0.373**

TABLE IV

COMPARISON OF POLICIES AMONG 3 ARCHITECTURES

Method \ Task	LiftPegUpright-v1	PokeCube-v1	PullCube-v1	PushCube-v1	RollBall-v1	Mean SR (↑)
DP (single-obs-step)	$\mathbf{0.645^{\pm 0.036}}$	$\underline{0.643}^{\pm 0.042}$	$\underline{0.783}^{\pm 0.018}$	$0.751^{\pm 0.026}$	$\mathbf{0.143^{\pm 0.023}}$	$\underline{0.593}$
DP (multi-obs-steps)	$0.0^{\pm 0.0}$	$0.06^{\pm 0.0}$	$0.1^{\pm 0.0008}$	$\mathbf{0.887^{\pm 0.0006}}$	$0.0^{\pm 0.0}$	0.209
DP (two-stream)	$\underline{0.601}^{\pm 0.038}$	$\mathbf{0.673^{\pm 0.022}}$	$\mathbf{0.863^{\pm 0.015}}$	$\underline{0.818}^{\pm 0.052}$	$\underline{0.139}^{\pm 0.020}$	**0.619**
FP (single-obs-step)	$\mathbf{0.729^{\pm 0.048}}$	$\mathbf{0.645^{\pm 0.036}}$	$\underline{0.771}^{\pm 0.027}$	$0.658^{\pm 0.037}$	$\underline{0.119}^{\pm 0.028}$	$\underline{0.582}$
FP (multi-obs-steps)	$0.0^{\pm 0.0}$	$0.049^{\pm 0.016}$	$0.040^{\pm 0.014}$	$\mathbf{0.860^{\pm 0.034}}$	$0.006^{\pm 0.005}$	0.191
FP (two-stream)	$\underline{0.634}^{\pm 0.026}$	$\underline{0.636}^{\pm 0.018}$	$\mathbf{0.775^{\pm 0.055}}$	$\underline{0.857}^{\pm 0.039}$	$\mathbf{0.184^{\pm 0.009}}$	**0.617**

TABLE V

COMPARATION BETWEEN DIFFERENT TEMPORAL FUSION METHODS

Method \ Task	LiftPegUpright-v1	PokeCube-v1	PullCube-v1	PushCube-v1	RollBall-v1	Mean SR (↑)
TempoFormer→concatenate	$0.634^{\pm 0.026}$	$\underline{0.636}^{\pm 0.018}$	$0.775^{\pm 0.055}$	$\mathbf{0.857^{\pm 0.039}}$	$\mathbf{0.184^{\pm 0.009}}$	$\underline{0.617}$
TempoFormer→average	$\underline{0.64}^{\pm 0.001}$	$0.6^{\pm 0.002}$	$\mathbf{0.846^{\pm 0.002}}$	$0.72^{\pm 0.004}$	$0.157^{\pm 0.001}$	0.593
cross-attention fusion	$0.571^{\pm 0.044}$	$0.610^{\pm 0.022}$	$0.794^{\pm 0.024}$	$0.698^{\pm 0.030}$	$0.170^{\pm 0.035}$	0.569
MTP (ours)	$\mathbf{0.652^{\pm 0.04}}$	$\mathbf{0.655^{\pm 0.034}}$	$\underline{0.837}^{\pm 0.027}$	$\underline{0.838}^{\pm 0.031}$	$\underline{0.172}^{\pm 0.036}$	**0.631**

is employed as the fusion module to merge the features from both the spatial and temporal streams.

The result in Tab. IV shows that DP (multi-obs-steps) / FP (multi-obs-steps) fail on four of the five tasks. The dramatic drop in performance compared to the normal DP (single-obs-steps) / FP (single-obs-steps) indicates that it's hard to learn to implicitly extract motion dynamics from multi-step observations. It is observed that the two-stream architecture uses the same fusion method as the multi-obs-steps architecture, but DP (two-stream) / FP (two-stream) achieve the best performance, which demonstrates that the two-stream architecture can improve the network's learning ability for multi-step inputs.

2) Ablation Experiments on Optical Flow Steps: We conducted an ablation study to evaluate the impact of the length of historical optical flow information on MTP performance. As shown in the Fig. 5, the model's success rate steadily improves with the number of optical flow steps, from 59.4% at 2 steps to 63.1% at 8 steps, significantly outperforming the baseline method at all test points. This trend confirms that longer motion histories provide the policy network with richer dynamic context, enabling better decision-making.

3) Ablation Experiments on Temporal Fusion Methods: We investigate the impact of different fusion modules in the temporal stream. Our MTP uses the TempoFormer in temporal stream to fuse the optical flow features. In our ablation

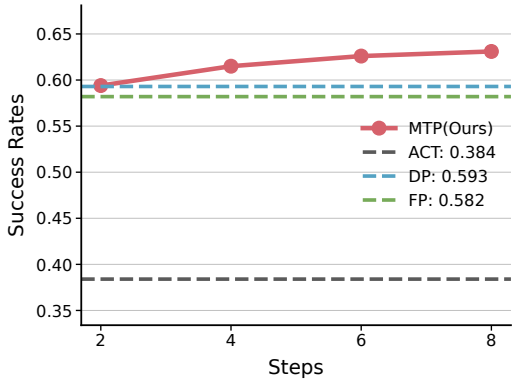

Fig. 5. Ablation Experiments on Optical Flow Steps

experiment for **TempoFormer→concatenate**, we replace the attention-based TempoFormer with a simple concatenation to fuse the features. In **TempoFormer→average** experiment, we use weighted averaging to fuse the temporal features and the weight $\mathbf{w} \in \mathbb{R}^T$ is learnable.

$$\begin{aligned} f^{opt} &= [f_1^{opt}, f_2^{opt}, \ldots, f_T^{opt}]\mathbf{w} \\ &= w_1 f_1^{opt} + w_2 f_2^{opt} + \cdots + w_T f_T^{opt} \end{aligned} \tag{5}$$

In **cross-attention fusion** experiment, we leverage cross-attention mechanisms that integrate spatial image features

optical flow features for fusion.

As the result shown at Tab. V, our method achieves the best performance on two tasks and ranks second on the remaining tasks, with only minor degradation compared to the highest success rates. These consistently high success rates demonstrate the effectiveness of our temporal fusion approach. The degradation at **cross-attention fusion** experiment demonstrates that our TempoFormer module can effectively aggregate temporal features to avoid spatial features being dominated.

V. Real World Experiments

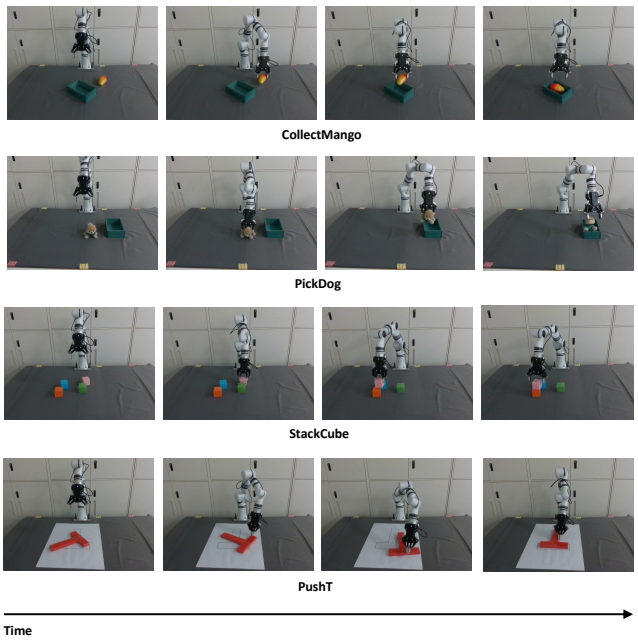

Fig. 6. We demonstrate ACT, Diffusion Policy, FlowMatching Policy and MTP on a real robotic setup. We evaluate on four tasks: Collect Mango, Pick Dog, Stack Cubes and PushT.

A. Setup

Platform. Our real-world experiments utilized a Realman robotic arm equipped with a DH AG95 gripper. A third person perspective mounted Intel RealSense D435i RGB-D camera provided a global RGB-D perspective of the workspace. All hardware components interfaced with a workstation equipped with an NVIDIA RTX 3090 GPU.

Demonstrations. Our experiments utilized 50 human-teleoperated demonstrations per task, gathered via Nintendo Switch Pro Controller. Each demonstration was carefully selected to reflect key skills and task-relevant interactions. This curated approach ensures a manageable yet representative dataset, reflecting the inherent complexities and challenges of each task.

Tasks. Our experiments encompass 4 primary tasks designed to evaluate the versatility and effec- tiveness of our proposed method in various manipulation scenarios: CollectMango, PickDog, StackCube and PushT. Each task was designed to test different aspects of robotic manipulation, including precision, adaptability, and real-time decision-making.

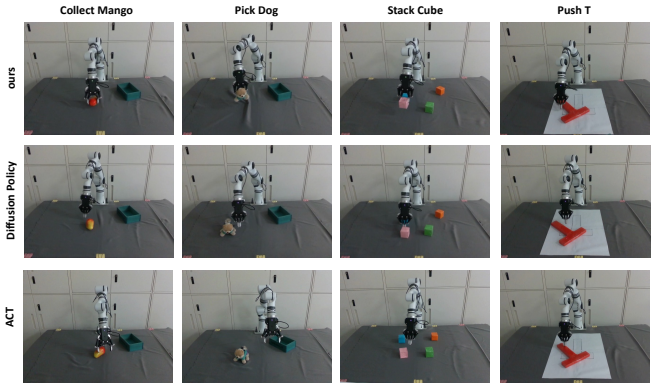

Fig. 7. Real World Experiment Examples

B. Result

TABLE VI
Real World Experiment

Task	CollectMango	PickDog	StackCube	PushT
ACT [6]	0.433	0.367	0.2	0.267
DP [7]	0.5	0.567	0.267	0.367
MTP (ours)	**0.7**	**0.633**	**0.4**	**0.533**

As shown in Tab. VI, the proposed method consistently achieves the best performance across all four tasks. For each task, we performed 30 trials with different initial states for the objects. In real-world experiments, MTP consistently demonstrated superior spatial perception compared to baselines. As the situation shown at Fig. 7, Diffusion Policy frequently struggles to localize the manipulated object accurately, resulting in task failure, while MTP consistently achieves precise object localization. Notably, the spatial stream in MTP, which perceives the object's position, shares the same core architecture as the encoder in Diffusion Policy. The improved spatial perception ability of MTP suggests that the incorporation of explicit motion context from temporal stream can guide the spatial stream to learn more effective representations. A video in the supplementary material will show failure cases to illustrate this conclusion.

VI. Conclusion

This paper introduces MTP, an imitation learning method that improves policy robustness by augmenting visuomotor learning with explicit motion priors from optical flow. With the explicitly motion information encoding, our method can narrow the gap between visual input and action output and improve the robustness for visual changes. We also introduces a two-stream architecture that separately processes RGB and optical flow inputs, enabling improved spatial-temporal understanding of dynamic environments. Extensive experiments in both simulated and real-world scenarios

demonstrate that MTP achieves state-of-art performance and is more robust to visual variation compared to other methods. In addition, the stronger spatial perception ability of MTP demonstrated at real world experiment indicates that the incorporation of motion priors derived from optical flow can direct the training of the spatial stream and improve spatial perception ability.

VII. Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (No.62372491), the Guangdong S&T Programme (No.2025B0101130003), the Guangdong Basic and Applied Basic Research Foundation (2022B1515020103, 2023B1515120087), and the Shenzhen Science and Technology Program under Grant (ZDCY20250901100201002).

References

[1] 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.

[2] B. Lim, J. Kim, J. Kim, Y. Lee, and F. C. Park, "Equigraspflow: Se (3)-equivariant 6-dof grasp pose generative flows," in *8th Annual Conference on Robot Learning*, 2024.

[3] N. M. M. Shafiullah, A. Rai, H. Etukuru, Y. Liu, I. Misra, S. Chintala, and L. Pinto, "On bringing robots home," *arXiv preprint arXiv:2311.16098*, 2023.

[4] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," *arXiv preprint arXiv:2401.02117*, 2024.

[5] S. Haldar, J. Pari, A. Rai, and L. Pinto, "Teach a robot to fish: Versatile imitation from one minute of demonstrations," *arXiv preprint arXiv:2303.01497*, 2023.

[6] 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.

[7] 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*, p. 02783649241273668, 2023.

[8] V. Vosylius, Y. Seo, J. Uruç, and S. James, "Render and diffuse: Aligning image and action spaces for diffusion-based behaviour cloning," *arXiv preprint arXiv:2405.18196*, 2024.

[9] P. Sundaresan, Q. Vuong, J. Gu, P. Xu, T. Xiao, S. Kirmani, T. Yu, M. Stark, A. Jain, K. Hausman, *et al.*, "Rt-sketch: Goal-conditioned imitation learning from hand-drawn sketches," in *8th Annual Conference on Robot Learning*, 2024.

[10] L.-H. Lin, Y. Cui, A. Xie, T. Hua, and D. Sadigh, "Flowretrieval: Flow-guided data retrieval for few-shot imitation learning," *arXiv preprint arXiv:2408.16944*, 2024.

[11] S. Xia, H. Fang, C. Lu, and H.-S. Fang, "Cage: Causal attention enables data-efficient generalizable robotic manipulation," *arXiv preprint arXiv:2410.14974*, 2024.

[12] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[13] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[14] E. Chisari, N. Heppert, M. Argus, T. Welschehold, T. Brox, and A. Valada, "Learning robotic manipulation policies from point clouds with conditional flow matching," *arXiv preprint arXiv:2409.07343*, 2024.

[15] 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.

[16] S. Lee, Y. Wang, H. Etukuru, H. J. Kim, N. M. M. Shafiullah, and L. Pinto, "Behavior generation with latent actions," *arXiv preprint arXiv:2403.03181*, 2024.

[17] T. Gervet, Z. Xian, N. Gkanatsios, and K. Fragkiadaki, "Act3d: 3d feature field transformers for multi-task robotic manipulation," *arXiv preprint arXiv:2306.17817*, 2023.

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

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

[20] H.-S. Fang, H. Fang, Z. Tang, J. Liu, C. Wang, J. Wang, H. Zhu, and C. Lu, "Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 653–660.

[21] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, *et al.*, "Droid: A large-scale in-the-wild robot manipulation dataset," *arXiv preprint arXiv:2403.12945*, 2024.

[22] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in neural information processing systems*, vol. 28, 2015.

[23] 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.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

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

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

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

[29] X. Liu, C. Gong, and Q. Liu, "Flow straight and fast: Learning to generate and transfer data with rectified flow," *arXiv preprint arXiv:2209.03003*, 2022.

[30] M. S. Albergo and E. Vanden-Eijnden, "Building normalizing flows with stochastic interpolants," *arXiv preprint arXiv:2209.15571*, 2022.

[31] Y. Li, W. H. Leng, Y. Fang, B. Eisner, and D. Held, "Flowbothd: History-aware diffuser handling ambiguities in articulated objects manipulation," *arXiv preprint arXiv:2410.07078*, 2024.

[32] C. Yuan, C. Wen, T. Zhang, and Y. Gao, "General flow as foundation affordance for scalable robot learning," *arXiv preprint arXiv:2401.11439*, 2024.

[33] M. Xu, Z. Xu, Y. Xu, C. Chi, G. Wetzstein, M. Veloso, and S. Song, "Flow as the cross-domain manipulation interface," *arXiv preprint arXiv:2407.15208*, 2024.

[34] T. Weng, S. M. Bajracharya, Y. Wang, K. Agrawal, and D. Held, "Fabricflownet: Bimanual cloth manipulation with a flow-based policy," in *Conference on Robot Learning*. PMLR, 2022, pp. 192–202.

[35] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime tv-l 1 optical flow," in *Pattern Recognition: 29th DAGM Symposium, Heidelberg, Germany, September 12-14, 2007. Proceedings 29*. Springer, 2007, pp. 214–223.

[36] Y. Wu and K. He, "Group normalization," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[37] 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.

[38] S. Tao, F. Xiang, A. Shukla, Y. Qin, X. Hinrichsen, X. Yuan, C. Bao, X. Lin, Y. Liu, T.-k. Chan, *et al.*, "Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai," *arXiv preprint arXiv:2410.00425*, 2024.

[39] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.