Imitation Learning Policy based on Multi-Step Consistent Integration Shortcut Model

Yu Fang, Xinyu Wang, Xuehe Zhang,*Member, IEEE*, Wanli Xue, Mingwei Zhang, Shengyong Chen,*Senior Member, IEEE*, Jie Zhao,*Senior Member, IEEE*

***Abstract*—The wide application of flow-matching methods has greatly promoted the development of robot imitation learning. However, these methods all face the problem of high inference time. To address this issue, researchers have proposed distillation methods and consistency methods, but the performance of these methods still struggles to compete with that of the original diffusion models and flow-matching models. In this article, we propose a one-step shortcut method with multi-step integration for robot imitation learning. To balance the inference speed and performance, we extend the multi-step consistency loss on the basis of the shortcut model, split the one-step loss into multi-step losses, and improve the performance of one-step inference. Secondly, to solve the problem of unstable optimization of the multi-step loss and the original flow-matching loss, we propose an adaptive gradient allocation method to enhance the stability of the learning process. Finally, we evaluate the proposed method in two simulation benchmarks and five real-world environment tasks. The experimental results verify the effectiveness of the proposed algorithm.**

***Index Terms*—Flow Matching, Shortcut, Consistency Model, Robot Learning**

I. Introduction

IMITATION learning (IL) is a fundamental paradigm for enabling robots to acquire motor skills by mapping sensory inputs—such as RGB images, depth maps, and proprioceptive signals—to appropriate actions. It has been widely adopted in diverse domains including impedance control [1] ,locomotion [2], grasping [3] and manipulation [4]–[6].

With advances in machine learning in Computer Vision [7]–[9], Natural Language Processing [10], and more, IL has evolved from simple toy demonstrations to complex, real-world robotic tasks. Recently, diffusion models have emerged as powerful generative backbones in IL, achieving remarkable performance and robustness. However, their iterative denoising process incurs high computational cost, making real-time deployment on physical robots impractical—especially when on-device compute budgets are tight. Therefore, there is an urgent need for efficient one-step policies that preserve the advantages of diffusion-based methods while drastically reducing computational overhead.

To address this issue, robotics researchers have borrowed knowledge-distillation techniques from computer vision to train efficient skill policies. Yet, a fundamental limitation persists: the student's performance is inherently bounded by that of the teacher, forming a practical upper ceiling. Unlike classic knowledge-distillation, recently the Shortcut framework [11] reformulates policy distillation into an online training paradigm that enforces a one-step policy and introduces an expected step-size objective.

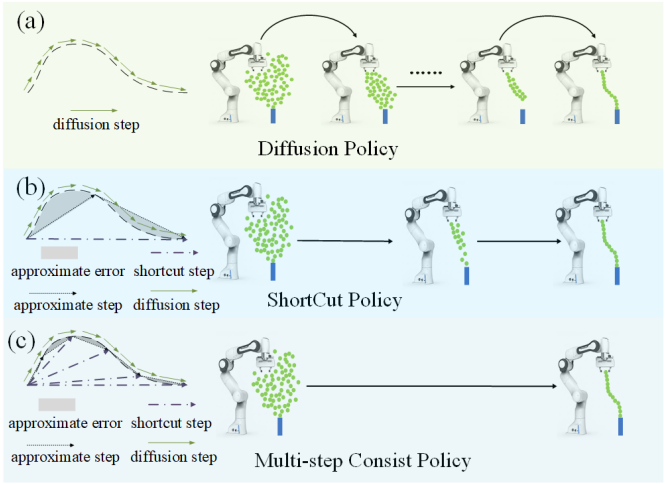


Fig. 1: Overview of the differences among diffusion models, Shortcut models, and our approach.

Although Shortcut demonstrates promising efficiency for robotic imitation learning, its one-step generation quality remains unsatisfactory. Shortcut can be interpreted as a hybrid between learning and distillation, where a reduced-step policy is distilled from the original flow-matching model. This coupling mode increases optimization difficulty; therefore, alleviating the learning burden becomes crucial for improving overall performance.

To overcome this challenge, a multi-step consistency strategy is introduced to facilitate optimization by supervising the policy with multiple-step predictions in this paper. Specifically, integration is performed along the same trajectory over several short steps, decomposing the long-range shortcut into a sequence of local approximations. This design reduces cumulative approximation errors and establishes multiple constraint points along the shortcut path, effectively distributing the one-step error across multiple intervals and improving the quality

This work was supported in part by the National Natural Science Foundation of China under Grant 62373129, Grant 92048301, and in part by the Heilongjiang Provincial Natural Science Foundation of China Grant YQ2023F012. (Yu Fang and Xinyu Wang contribute equally to this work, Corresponding authors: Xuehe Zhang, Jie Zhao)

Yu Fang, Xinyu Wang, Xuehe Zhang, Mingwei Zhang, and Jie Zhao are with the State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, Harbin 150001, China. (e-mail: 20b308008@stu.hit.edu.cn,wxinyu@hit.edu.cn, zhangxuehe@hit.edu.cn,21B908040@stu.hit.edu.cn,jzhao@hit.edu.cn).

Wangli Xue and Shengyong Chen are the School of Computer Science and Engineering, Tianjin University of Technology, Tianjin 300384, China. (e-mail: xuewanli@email.tjut.edu.cn,sy@ieee.org).

of one-step generation.

Figure 1 conceptually compares our method with prior approaches. As shown, diffusion models rely on iterative denoising with small steps, while Shortcut introduces an expected step length to generate motion policies with fewer iterations. Our approach further extends Shortcut by averaging one-step losses across multiple steps, improving generation speed without compromising quality. The standard Shortcut configuration can be viewed as a special case of our framework when the step number is set to two.

Because the multi-step consistency loss measures self-consistency, its magnitude is typically small—often close to zero at the beginning of training—while the flow-matching loss remains considerably larger. Consequently, the gradient signal from the consistency term may be easily dominated. To maintain a balanced contribution between the two objectives, we introduce a gradient-adaptive scheme inspired by multi-task gradient correction. This method reorients the composite gradient to mitigate imbalance, ease optimization, and improve overall performance.

In summary, to alleviate the excessive iterative steps in diffusion-based imitation learning, we propose a multi-step consistency framework that markedly improves one-step generation quality and facilitates the practical deployment of diffusion models in robot policy learning. The key contributions are as follows:

1) Introduce a multi-step consistency imitation-learning algorithm that enhances one-step generation by supervising with multi-step predictions.
2) Develop a gradient-adaptive optimization scheme to balance the flow-matching and consistency objectives, preventing domination by any single loss and improving inference performance.
3) Demonstrate the effectiveness of the proposed framework on robotic manipulation, highlighting its potential as a general-purpose imitation learning approach.

The remainder of this paper is organized as follows. Section 2 reviews related work, Section 3 presents the proposed method, Section 4 reports experimental results, and Section 5 concludes the paper.

II. Related works

A. Diffusion Model For Robotics

Numerous studies have applied diffusion models, flow matching, and related generative techniques to robot skill learning. Diffusion Policy [12] significantly improves skill-learning performance on diverse manipulation tasks by introducing action chunking and other training refinements. To deploy Diffusion Policy on tasks driven by 3D point-cloud perception, researchers improve computational efficiency by simplifying point-cloud network architectures [13]. Furthermore, distillation-based approaches have been explored for skill learning [14].

Beyond diffusion models, the concise formulation of flow matching [15] has motivated its adoption in robot skill learning, where it has shown strong performance [16]. Unlike prior work, our method targets strong one-step inference from a single training procedure. Because training occurs once (without teacher models), performance is independent of distillation and does not rely on external teachers, avoiding teacher-limited accuracy. Moreover, our approach is orthogonal to point-cloud backbone design; improvements in point-cloud networks can be seamlessly integrated into our framework.

B. One-step Inference in Diffusion Model

To improve real-time inference with diffusion models, researchers have analyzed their theoretical properties and proposed a range of sampling-acceleration methods. Examples include DDIM [17], EDM [18], and DPM-Solver [19], as well as rectified flow matching. In parallel, knowledge distillation compresses multi-step diffusion into fewer steps—sometimes even a one-step. For the distillation objective, alternatives to L2 loss have been explored, including adversarial [20] and distribution-matching [21] objectives.

Closely related to our work is Shortcut [11]. Recently, researchers introduced an expected step length to enable end-to-end training of one-step generation, a formulation that can be viewed as self-distillation. To mitigate sampling error in Shortcut, researchers proposed higher-order Shortcut variants from an inference perspective [22]. One concurrent work with ours is Meanflow [23]. It obtains an analytical expression for the average velocity through back-propagation of the velocity gradient. However, this method has poor stability because it relies on the Jacobian Vector Product.

Unlike prior methods, our approach unifies training and inference within a single framework. We perform multi-step integrations along the same trajectory and average per-step integration errors across steps. This procedure yields high-quality one-step generation.

III. Method

Recently, many iterative generative methods—such as diffusion models and flow matching—have adopted ordinary differential equation (ODE) formulations to model the transition from a noise distribution to the data distribution. Owing to its concise formulation and the advantage of bypassing reverse-time SDE simulation, flow matching has attracted broad interest. In flow matching, $\mathbf{x}_t$ is defined as the interpolation between the noise distribution $\mathbf{x}_0$ and the data distribution $\mathbf{x}_1 \sim \mathcal{D}$, and $\mathbf{v}_t = \frac{d\mathbf{x}_t}{dt}$ is defined as the flow velocity from the noise distribution to the data distribution. In this paper, we adopt linear interpolation to define $\mathbf{x}_t$ and $\mathbf{v}_t$:

$$\begin{aligned} \mathbf{x}_t &= (1-t)\mathbf{x}_0 + t\mathbf{x}_1 \\ \mathbf{v}_t &= \mathbf{x}_1 - \mathbf{x}_0 \end{aligned} \tag{1}$$

Given noise-data pairs $\mathbf{x}_0$, $\mathbf{x}_1$, and the interpolation method, $\mathbf{v}_t$ can be determined. However, if $\mathbf{x}_1$ is known, there is no need to calculate $\mathbf{v}_t$. From the perspective of an ordinary differential equation (ODE), since $\mathbf{v}_t = \frac{d\mathbf{x}_t}{dt}$, $\mathbf{v}_t$ can be solved by specifying $\mathbf{x}_t$. Thus, the flow model can be optimized by regressing the true velocity of the randomly sampled noise-data pairs $(\mathbf{x}_0, \mathbf{x}_1) \sim \mathcal{D}$:

$$\mathcal{L}_{\mathrm{F}}(\theta) = \mathbb{E}_{\mathbf{x}_0,\mathbf{x}_1\sim D}\left[\left\|\bar{\mathbf{v}}_\theta\left(\mathbf{x}_t, t\right) - \left(\mathbf{x}_1 - \mathbf{x}_0\right)\right\|^2\right] \tag{2}$$

After training, the learned velocity field $\bar{\mathbf{v}}_\theta(\mathbf{x}_t, t)$ is used to evolve a sample drawn from noise $\mathbf{x}_0$ toward a target sample $\mathbf{x}_1$. Sampling proceeds via numerical integration of the ODE, i.e., iterative updates of $\mathbf{x}_t$ with a chosen step size.

$$\begin{aligned}\mathbf{x}_1 &= \mathbf{x}_0 + \int_{t=0}^{t=1} \mathbf{v}\left(\mathbf{x}_t, t\right) dt \\ &= \mathbf{x}_0 + \sum_{t=0}^{t=1} \mathbf{v}\left(\mathbf{x}_t, t\right) \delta t\end{aligned} \tag{3}$$

In ODE solvers, the step size is pivotal: smaller steps usually yield better discretization accuracy but require more iterations. To mitigate this cost, Shortcut augments the flow model with an explicit step-size parameter, replacing the original output $\mathbf{v}_t(\mathbf{x}_t, t)$ with $\mathbf{v}_t(\mathbf{x}_t, t, d)$, where d denotes the step size. To train this parameterization, Shortcut employs a self-consistency objective:

$$\mathbf{v}(\mathbf{x}_t, t, 2d) = \frac{\mathbf{v}(\mathbf{x}_t, t, d) + \mathbf{v}(\hat{\mathbf{x}}_{t+d}, t+d, d)}{2} \tag{4}$$

By incorporating this self-consistency objective into the original flow-matching objective, the entire training objective for ShortCut becomes

$$\begin{aligned}\mathcal{L}_{\mathrm{S}}(\theta) = \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_1 \sim D} \Big[&\left\| \bar{\mathbf{v}}_\theta\left(\mathbf{x}_t, t, 0\right) - \left(\mathbf{x}_1 - \mathbf{x}_0\right) \right\|^2 \\ &+ \left\| \bar{\mathbf{v}}_\theta\left(\mathbf{x}_t, t, 2d\right) - \mathbf{v}(\mathbf{x}_t, t, 2d) \right\|^2 \Big]\end{aligned} \tag{5}$$

The sampling process also changes:

$$\mathbf{x}_1 = \mathbf{x}_0 + \sum_{t=0}^{t=1} \mathbf{v}\left(\mathbf{x}_t, d, t\right) d \tag{6}$$

We analyze Equation 6 and find that v_t is no longer the instantaneous velocity at time t, but the average velocity between $(t, t+d)$. Therefore, ShortCut essentially unifies the objectives of the inference and training processes. However, since the self-consistency loss term of ShortCut is based on two small step sizes, their accuracy determines the accuracy of larger step sizes. When the step size is too large, the extrapolation error becomes too large, making it difficult for ShortCut to estimate accurately. Thus, an effective approach is to reduce the extrapolation error. Consequently, we consider a more general inference process on the same trajectory:

$$\begin{aligned}\mathbf{x}_{t+nd} =& \mathbf{x}_t + \mathbf{v}\left(\mathbf{x}_t, d, t\right) d + \mathbf{v}\left(\mathbf{x}_{t+d}, d, t+d\right) d \\ &+ \cdots + \mathbf{v}\left(\mathbf{x}_{t+(n-1)d}, d, t+(n-1)d\right) d\end{aligned} \tag{7}$$

We can break down this process and write it as follows:

$$\begin{aligned}\hat{\mathbf{x}}_{t+nd} &= \hat{\mathbf{x}}_{t+(n-1)d} + \mathbf{v}\left(\hat{\mathbf{x}}_{t+(n-1)d}, d, t+(n-1)d\right) d \\ \hat{\mathbf{x}}_{t+(n-1)d} &= \hat{\mathbf{x}}_{t+(n-2)d} + \mathbf{v}\left(\hat{\mathbf{x}}_{t+(n-2)d}, d, t+(n-2)d\right) d \\ &\cdots \\ \hat{\mathbf{x}}_{t+2d} &= \hat{\mathbf{x}}_{t+d} + \mathbf{v}\left(\hat{\mathbf{x}}_{t+d}, d, t+d\right) d \\ \hat{\mathbf{x}}_{t+d} &= \mathbf{x}_t + \mathbf{v}\left(\mathbf{x}_t, d, t\right) d\end{aligned} \tag{8}$$

From Equation 8, the integration over a long horizon can be decomposed into multiple shorter integrations, each reusing the previous integration terms. Consequently, for n integration points along the same trajectory, we obtain $(n-1)$ consistency constraints. Because each segment is constrained, the per-segment extrapolation error is also constrained. We formulate the training objective in the Shortcut style:

$$\begin{aligned}nd\mathbf{v}\left(\mathbf{x}_t, nd, t\right) = &\underbrace{d \cdot \mathbf{v}\left(\mathbf{x}_t, d, t\right) + d\mathbf{v}\left(\mathbf{x}_{t+d}, d, t+d\right)}_{2d\mathbf{v}(\mathbf{x}_t, 2d, t)} + \cdots \\ &\underbrace{+ d\mathbf{v}\left(\mathbf{x}_{t+(n-2)d}, d, t+(n-2)d\right)}_{(n-1)d\mathbf{v}(\mathbf{x}_t, (n-1)d, t)} \\ &+ d\mathbf{v}\left(\mathbf{x}_{t+(n-1)d}, d, t+(n-1)d\right)\end{aligned} \tag{9}$$

Our loss function consists of two parts :

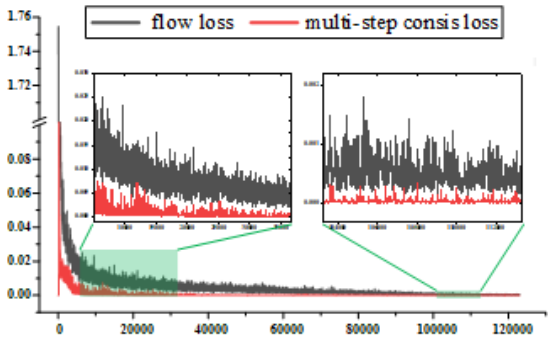
$$\begin{aligned}\mathcal{L}_{\mathrm{M}}(\theta) = \{\mathbb{E}_{\mathbf{x}_0, \mathbf{x}_1 \sim D} &\underbrace{\left[\left\| \bar{\mathbf{v}}_\theta\left(\mathbf{x}_t, t, 0\right) - \left(\mathbf{x}_1 - \mathbf{x}_0\right) \right\|^2 \right.}_{\text{Flow Matching Loss:}\mathcal{L}_{FM}} \\ &\underbrace{\left. , \sum_{k=2}^{n} \left\| \bar{\mathbf{v}}_\theta\left(\mathbf{x}_t, t, kd\right) - \mathbf{v}(\mathbf{x}_t, t, kd) \right\|^2 \right] \}}_{\text{Multi Consis Loss:}\mathcal{L}_{MC}}\end{aligned} \tag{10}$$

The training procedure of our algorithm is summarized in the Algorithm 1.

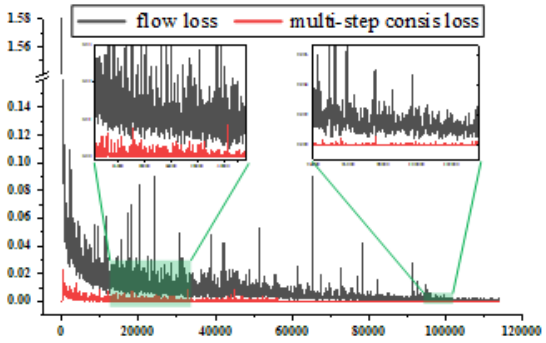
In the robot imitation-learning setting, our goal is to map noise $\mathbf{a}_0$ to a target action distribution $\mathbf{a}_1$. Accordingly, we define a conditioned action-generation velocity field $\bar{\mathbf{v}}_\theta(\mathbf{a_t}, \mathbf{t}, \mathbf{d}, \mathbf{o})$. Given demonstrations $\{\mathbf{o}, \mathbf{a}_1\}$ and noise samples $\mathbf{a}_0$, our training objective comprises two components:

Algorithm 1 Multi-Step Consistency Flow Matching Training

while not converged **do**
  $\mathbf{x}_0 \sim \mathcal{N}(0, \mathcal{I}), \mathbf{x}_1 \sim D, t \sim p(t), d \sim p(d), c, k,$ **v_list**
  $\mathbf{x}_t = t\mathbf{x}_0 + (1-t)\mathbf{x}_1$
  for first m samples **do**
    $\mathbf{v}_t = \mathbf{x}_1 - \mathbf{x}_0$
    $d \longleftarrow 0$
  end for
  for other samples **do**
    for n in range k-1 **do**
      if n==0 **then**
        v_list.append$(\hat{\mathbf{v}}_\theta(\mathbf{x}_t, t, d))$
        $\hat{\mathbf{x}}_{t+d} \longleftarrow \mathbf{x}_t + \hat{\mathbf{v}}_\theta(\mathbf{x}_t, t, d)d$
      else
        v_list.append$(\hat{\mathbf{v}}_\theta(\hat{\mathbf{x}}_{t+(n-1)d}, t+(n-1)d, d))$
        $\hat{\mathbf{x}}_{t+nd} \longleftarrow \hat{\mathbf{x}}_{t+(n-1)d} + \hat{\mathbf{v}}_\theta(\hat{\mathbf{x}}_{t+(n-1)d}, t+(n-1)d, d)d$
      end if
    end for
  end for
  for n in range(2,k+1) **do**
    $d \longleftarrow nd$
    $\mathbf{v}_t \longleftarrow \frac{1}{m}\sum_{i=0}^{m}$ v_list$[i]$
  end for
  $\theta \longleftarrow \nabla_\theta \|\bar{\mathbf{v}}_\theta(\mathbf{x_t}, \mathbf{t}, \mathbf{d}) - \mathbf{v}_t\|^2$
end while



(a) block hammer beat



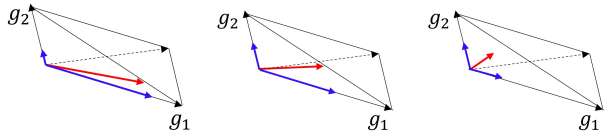
(b) bottle adjust

Fig. 2: The difference between the two losses is too large in value.

$$\mathcal{L}_{\text{IL}}(\theta) = \{\mathbb{E}_{\mathbf{a}_0,\mathbf{a}_1,\mathbf{o}\sim D}\Big[\|\bar{\mathbf{v}}_\theta\left(\mathbf{a}_t,\mathbf{o},t,0\right)-(\mathbf{a}_1-\mathbf{a}_0)\|^2 \\ ,\sum_{k=2}^{n}\|\bar{\mathbf{v}}_\theta\left(\mathbf{a}_t,\mathbf{o},t,kd\right)-\mathbf{v}(\mathbf{a}_t,\mathbf{o},t,kd)\|^2\Big]\} \tag{11}$$

In robot imitation learning, low-dimensional action spaces exhibit weaker distributional structure than images, making it harder for networks to discriminate noise from action trajectory samples. Moreover, because the multi-step consistency objective depends on the flow-matching objective, naively mixing multi-step consistency and flow-matching gradients during joint training induces gradient imbalance. Specifically, the multi-step consistency loss is typically smaller in scale, as shown in Figure 2, so the flow-matching loss dominates optimization, leaving the multi-step consistency objective undertrained and degrading one-step performance. In our setting, internal dependencies within the multi-step objective, combined with even lower-dimensional actions, further exacerbate training difficulty.

To address this issue, we cast the integrated objective as a multi-task learning (MTL) problem. Building on established MTL methods—PCGrad [24], dynamic weight averaging (DWA) [25], and IMTL [26]—we propose a new Adaptive Gradient Allocation (AGA) algorithm. Although the multi-step consistency and flow-matching objectives are related, their dependence is asymmetric: the multi-step consistency loss depends on the flow-matching loss, not vice versa. This



(a) (b) (c)

Fig. 3: By adjusting the scaling factor c, we can control the projection intensity of the composite gradient in different directions. Red arrows denote the composite gradient; blue arrows denote its projections onto different directions.

asymmetry makes standard MTL schemes suboptimal for this configuration, motivating our AGA design.

The proposed AGA introduces an adaptive gradient-composition mechanism that adjusts the gradient direction as learning progresses across the two interdependent tasks. As shown in Fig. 3, Figs. 3a–3c illustrate that the composite-gradient direction gradually shifts from $\mathbf{u}_1$ toward $\mathbf{u}_2$. Specifically, the projections of the composite gradient onto the respective task subspaces must satisfy the following criteria:

$$c(\mathbf{g}\mathbf{u}_1^\top) = \mathbf{g}\mathbf{u}_2^\top \tag{12}$$

Let $\mathbf{u}_1 \in \mathbb{R}^d$ denote the unit gradient direction induced by the flow-matching loss, and $\mathbf{u}_2 \in \mathbb{R}^d$ the corresponding unit direction from the multi-step consistency loss. The composite gradient vector is formulated as $\mathbf{g} = \alpha_1\mathbf{g}_1 + \alpha_2\mathbf{g}_2$, where $\mathbf{g}_1 = A\mathbf{u}_1$ and $\mathbf{g}_2 = B\mathbf{u}_2$ correspond to the gradient of the flow-matching loss and multi-step consistency loss, respectively. Here, $A = \|\nabla_\theta \mathcal{L}_{\text{FM}}\|_2$ and $B = \|\nabla_\theta \mathcal{L}_{\text{MC}}\|_2$ represent the L_2-norm magnitudes of their respective gradient vectors.

Then we can obtain the closed-form solution:

$$\alpha_1 = \frac{B(1-c\delta)}{A(c-\delta)+B(1-c\delta)}, \quad \alpha_2 = 1-\alpha_1 \tag{13}$$

Here, δ is the cosine of the angle between the two gradient vectors, and the detailed derivation process is presented in Appendix A.

Based on equation 12, we can specify c to adjust the strength of the composite gradient in the directions of the two tasks. To ensure that the tasks can be automatically adjusted according to the actual situation, we introduce the loss descent rate $v_i = \frac{\mathcal{L}_i^{\text{current}}}{\mathcal{L}_i^{\text{last}}}$ for regulation. We aim to increase the projection in the direction of the gradient generated by the multi-step consistency loss when its descent rate is relatively slower than that of the flow-matching loss (i.e.,v_2 is greater, indicating a slower descent). Thus, we introduce the intensity adaptive adjustment strategy:

$$c_{new} = c\exp\left(\gamma(\frac{v_2}{v_1}-1)\right) \tag{14}$$

Finally, we use Exponential Moving Average (EMA) to adjust c:

$$c = \beta c + (1-\beta)c_{new} \tag{15}$$

We can derive the constraint relationship between c and the gradients based on the constraints $\alpha_1 \in (0,1)$. If the condition

Algorithm 2 Adaptive Gradient Allocation

Input $\theta, \{\mathcal{L}_{FM}, \mathcal{L}_{MC}\}, \beta$
$\mathbf{g}_1 \longleftarrow \nabla_\theta \mathcal{L}_{FM}$, $\mathbf{g}_2 \longleftarrow \nabla_\theta \mathcal{L}_{MC}$
for k in range N_{step} **do**
  if $k < n_{start}$ **then**
    $\mathbf{g}_1^{bk} \longleftarrow \mathbf{g}_1$, $\mathbf{g}_2^{bk} \longleftarrow \mathbf{g}_2$
    if $\mathbf{g}_2 \cdot \mathbf{g}_1^{bk} < 0$ **then**
      $\mathbf{g}_1^{bk} = \mathbf{g}_1^{bk} - \frac{\mathbf{g}_2 \cdot \mathbf{g}_1^{bk}}{\|\mathbf{g}_2\|^2} \cdot \mathbf{g}_2$
    end if
    if $\mathbf{g}_1 \cdot \mathbf{g}_2^{bk} < 0$ **then**
      $\mathbf{g}_2^{bk} = \mathbf{g}_2^{bk} - \frac{\mathbf{g}_1 \cdot \mathbf{g}_2^{bk}}{\|\mathbf{g}_1\|^2} \cdot \mathbf{g}_1$
      $\alpha_1 = 0.5, \alpha = 0.5$
      $\mathbf{g}_1 \longleftarrow \mathbf{g}_1^{bk}$, $\mathbf{g}_2 \longleftarrow \mathbf{g}_2^{bk}$
    end if
  else
    compute c_{new} using Equation 14
    update c using Equation 15
    if c satisfies Equation 16 **then**
      compute α_1 and α_2 using 13
    end if
  else
    $\alpha_1 = 0.5, \alpha = 0.5$
  end if
  return $\Delta\theta = \alpha_1 \mathbf{g}_1 + \alpha_2 \mathbf{g}_2$
end for

is satisfied, we proceed with the adaptive adjustment. If not, we simply set $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$. The relationship that c should satisfy with the gradients is as follows:

$$\begin{cases} \delta < c < \dfrac{A\delta - B}{A - B\delta} & \textbf{if} \quad \delta > 0 \quad \textbf{and} A < B\delta \\ \max\left\{\delta, \dfrac{A}{B}\right\} < c & \textbf{if} \quad \delta > 0 \quad \textbf{and} A = B\delta \\ \max\left\{\delta, \dfrac{A\delta - B}{A - B\delta}\right\} < c & \textbf{if} \quad \delta > 0 \quad \textbf{and} A > B\delta \\ 0 < c & \textbf{if} \quad \delta \leq 0 \end{cases} \tag{16}$$

The detailed derivation process is presented in Appendix B.

In addition, we found that the multi-step consistency loss cannot provide effective guidance (usually 0) at the initial stage of the task. Therefore, in the early stage of the task, we directly employ the pcgrad [24] method. Overall, the detailed gradient adaptive algorithm is presented in Algorithm2.

Finally, the velocity field is solved using the previously proposed UDiT1d architecture in the [27]. The network architecture is shown in Figure 4. Unlike our earlier setup in the [27], this study does not employ self-supervised learning. Accordingly, the sliding decoder is removed, leaving a standard encoder–decoder architecture. All other components remain unchanged, including the Rotary Positional Encoding and addLLN.

During inference, occasional draws of low-probability samples can degrade performance. To mitigate this, inspired by the codebook in [28], we construct a prior codebook sampled from a Gaussian distribution. For both the multi-step consistency target and one-step inference, sampling is restricted to the codebook, thereby avoiding low-probability draws.

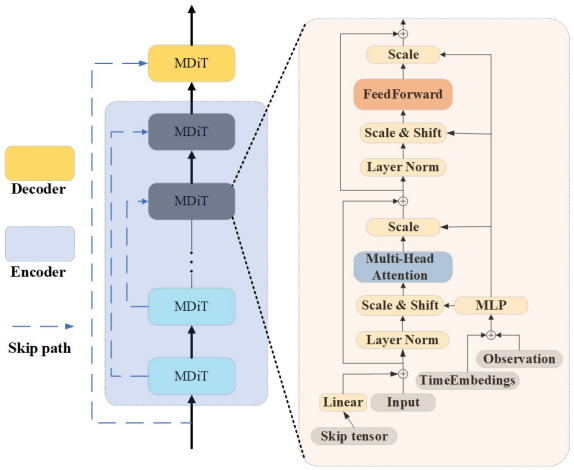


Fig. 4: UDiT1d architecture.

IV. Experiment

This section evaluates the proposed method across simulated and real-world tasks. In simulation, two benchmarks with point-cloud and state observations assess performance on standard and long-horizon manipulation. Similarly to the simulation environment, in the real world, we also evaluate multiple tasks to assess the performance of the proposed method on ordinary manipulation tasks and long-horizon manipulation tasks. In addition, in the real world, we evaluate the effectiveness of the method proposed in this paper on tasks with severe external disturbances. To probe high-frequency contact behavior, we additionally evaluate peg-in-hole and wiping tasks augmented with tactile sensing.

In summary, this section primarily addresses the following questions:

1) Can the proposed method solve general manipulation tasks and long-horizon manipulation tasks in the simulation environment?
2) How does the proposed method perform in real-world tasks?
3) Can the proposed method be generalized to multimodal fusion tasks?
4) Can the handling of gradients truly enhance the performance of the algorithm?

A. Simulation Environments

1) RoboTwin [29]: is a robot benchmark suite specifically designed for bimanual robot tool usage and human-robot interaction scenarios, generating diverse demonstration data through traditional planners, as shown in 5. We collected 50 demonstration trajectories for each task, using the point cloud as the observation.

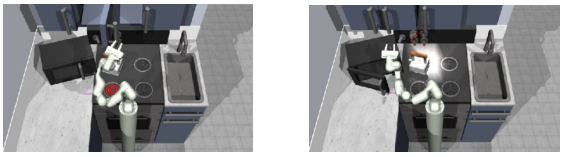
2) Franka Kitchen [30]: focuses on testing long sequence tasks. It uses states as observation information, rather than point clouds or images. It contains 7 interactive objects and a manual demonstration data set with 566 demonstrations, and

(a) block hammer beat (b) bottle adjust (c) container place (d) diverse bottles pick

(e) dual bottles pick (f) dual shoes place (g) mug hanging (h) pick apple messy

Fig. 5: The experiments in the RoboTwin benchmark.

each demonstration completes 4 tasks in any order, as shown in 8. The goal is to perform as many demonstration tasks as possible, regardless of the order, and display both short-term and long-term multimodal.



(a) (b)

Fig. 6: The experiments in the Franka Kitchen. a) The robot interacts with the kettle, burner, slider, and cabinet. b) The robot interacts with the kettle, microwave, top burner, and lights.

We selected Diffusion, Meanflow and Shortcut as baselines. For Diffusion, we used the UNet architecture as the noise prediction network, while others including Shortcut,Meanflow and our proposed method utilized the UDiT1d as the velocity prediction network. It should be noted that in Robotwin, we employed the 3D-diffusion policy (3DP), whereas in Franka Kitchen, we used the Diffusion Policy (DP). In Robotwin, following the evaluation protocol of Robotwin, we collected 50 successful demonstrations. Using 3D point clouds as observations, each algorithm was trained three times with three different random seeds for 3000 epochs. The last epoch's checkpoint was selected for evaluation, which was conducted 100 times to calculate the success rate. In the Franka Kitchen environment, we used the dataset provided by Diffusion Policy. Each algorithm was trained for 3000 epochs with three different random seeds. The last epoch's checkpoint was chosen for evaluation, which was performed 100 times to calculate the success rate of interactions with different objects.

The experimental results on the Robotwin benchmark are shown in Table I. Our method outperform the one-step Shortcut in all 10 tasks and Meanflow in 7 task, even surpass the 3DP with 10 iterations in 6 tasks. With an average success rate of 44.27% across the 10 tasks, our proposed method outperformed the 3DP's 41.39%, which fully demonstrates the strong competitiveness of the algorithm proposed in this paper. The experimental results on the Franka Kitchen benchmark are shown in Table II. In terms of object interaction, our method outperform the one-step Shortcut and Meanflow, it is comparable to the DP with 100 iterations, showing strong competitiveness. In summary, the experimental results in simulation indicate that the proposed method is highly competitive in various task scenarios.

TABLE I: Scores on simulated task with the point cloud observation on RoboTwin. We report an average of 300 episodes and a standard deviation of 3 seeds.

Task	3DP	Meanflow	ShortCut	Ours
block hammer beat	64.7±10.1	24±12.8	44.7±1.7	**81.3±0.9**
bottle adjust	**71.7**±13.8	62.3±9.7	59.3±6.1	59.0±15.7
diverse bottles pick	32.3±10.1	51.0±5.7	35.7±4.2	**55.0±1.4**
dual bottles pick(easy)	74.7±2.9	86.6±4.9	45.0±9.7	**89.3±0.5**
dual bottles pick(hard)	**48.0**±7.9	48±5.7	39.7±5.4	46.7±1.7
dual shoe place	7.7±2.1	11±4.3	9.4±4.1	**12.7±3.3**
pick apple messy	12.7±5.5	12.6±4.2	13.3±7.6	**13.7±9.2**
container place	**77.7±2.5**	65.6±4.5	65.6±2.5	55.0±5.7
mug hanging(easy)	**14.0±2.3**	7.3±4.6	2.6±7.6	13.3±4.1
mug hanging(hard)	10.7±3.1	7±5.7	1.7±1.2	**16.7±1.7**
NFE	10	1	1	1

TABLE II: Scores on Franka Kitchen with the state observation. We report an average of 300 episodes and a standard deviation of 3 seeds.

Task	DP	Meanflow	ShortCut	Ours
p1	**100±0.0**	**100±0.0**	99.3±0.5	**100±0.0**
p2	**100±0.0**	99±0.0	99.3±0.05	**100±0.0**
p3	99.7±0.5	98±0.8	98.3±1.2	**100±0.0**
p4	**98.7±0.5**	94±0.8	96.3±1.7	98.0±1.4
NFE	100	1	1	1

TABLE III: Scores on real-world task.

Task	DP	ShortCut	Ours
Object classification	40%	10%	**50%**
Meal packaging	30%	10%	**40%**
Dynamic placement	40%	20%	**50%**

B. Real-world Environments

We evaluated the effectiveness of the proposed method in this paper on tasks that utilize images as input for perception. We use the DP and the Shortcut as baselines. We selected three tasks: object classification and meal packaging, and dynamic placement tasks.

1) Object classification: In this task, the robot needs to learn to place different randomly placed objects into different designated areas, which is a very conventional test task. We used two cameras, the D455 and D405, for image observation. We employed Gello as the data acquisition tool and collected 81 complete and successful sets of experimental data, with a data acquisition frequency of 20 Hz.

2) Meal packaging: In this task, the robot needs to first open the lunch box and then place the corn and buns into it, with the positions of the lunch box, corn, and buns all being random. This is a long-horizon task similar to Franka Kitchen, and the operations have a specific sequence, as the lunch box must be opened first before any subsequent actions can be performed. Like in object classification, we used two cameras, the D455 and D405, for image observation and collected 92 complete and successful demonstration datasets, with a data acquisition frequency of 20 Hz.

3) Dynamic placement: In this task, the robot needs to place two different randomly placed objects into a designated box, with both the objects and the box being randomly positioned, and the box's position also being subject to human interference during task execution. This task is primarily designed to evaluate the algorithm's agility in quickly adapting to changes in the external environment. Similar to object classification, we used two cameras, the D455 and D405, for image observation, with a data acquisition frequency of 20 Hz, and collected 100 complete and successful demonstration datasets. There is no human interference in the collected demonstration datasets.

For these three tasks, we all used the AdamW optimizer, trained for 500 epochs, and set the learning rate to 1.0e-4. We utilized the weights from the last checkpoint to test the performance of the algorithm. For each task, we conducted evaluations 10 times and then calculated the average performance. Our experimental results are shown in Table III, which demonstrates that our method exhibits strong competitiveness across all three tasks. Specifically, our method outperforms Diffusion Policy in terms of performance and requires fewer NFE (number of function evaluations). The experimental videos can be viewed in Part I of the supplemental video.

C. Multimodal tasks in Real-world

To further explore whether the method proposed in this paper can be extended to multimodal-based robotic manipulation tasks, we conducted experiments on two tasks: peg-in-hole and wiping, which incorporate tactile perception. For the multimodal perception component, we used two ResNet-18 models to process visual and tactile information, respectively, and then employed a transformer to fuse these two types of information. For the skill learning component, we still adopt the network structure of UDiT1d. The detailed information on these tasks is as follows:

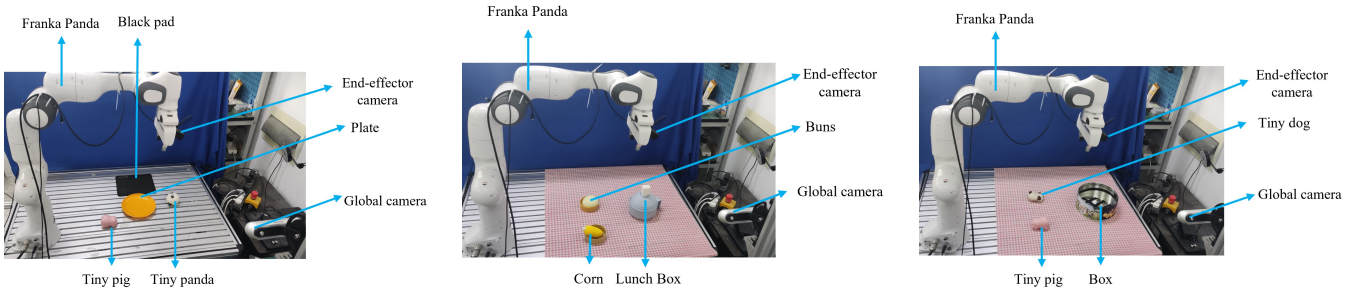
1) Peg-in-hole: The peg-in-hole task requires picking up a cylinder and inserting it into a base accurately, which is a common visual-tactile manipulation task. For the perception part, we use a camera mounted at the end of the robotic arm to capture the manipulation scene and a GelSight Mini installed on the gripper to sense the tactile information during the contact process. We collected 80 sets of successful and complete experimental data, with a data acquisition frequency of 20 Hz.

2) Wiping: In this task, the robot first needs to grasp the eraser in the working area and then erase the black lines on the whiteboard. This task is more difficult than the peg-in-hole task, as the robot is required to maintain prolonged contact with the whiteboard using an appropriate force. Similarly to the peg-in-hole task, in this task, we use a camera mounted at the end of the robotic arm to acquire visual information and a GelSight Mini installed on the gripper to obtain tactile information. We collected 106 sets of successful and complete demonstrations, with a data acquisition frequency of 20 Hz.

We evaluate the performance of the algorithm on the Peg-in-Hole task using the success rate and assess its performance on the Wiping task using the completion rate, which refers to the proportion of the erased area relative to the entire black line area. For each task, we conducted 10 tests in a real-world environment and then calculated the average performance; the experimental results are shown in Fig 9. Our method achieved a 40% success rate on the Peg-in-Hole task and a 47.8% completion rate on the Wiping task. The algorithm's performance on both tasks significantly outperformed that of the diffusion policy. We also found that the method using a shortcut substantially outperformed the diffusion policy. We attribute this to the algorithm's short inference time, which enables it to demonstrate excellent performance in complex interactive tasks. The experimental videos can be viewed in Part II of the supplemental video.

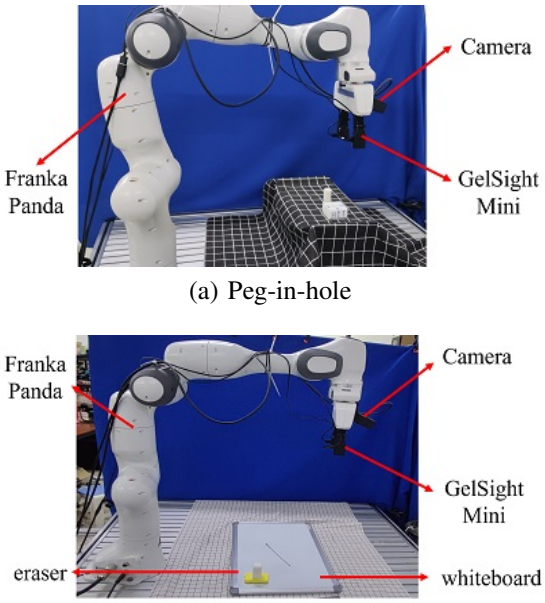
D. Ablation

To further investigate the influence of computing consistency losses with different step numbers during the training phase on the final performance of the algorithm. To control the proportion of data involved in the multi-step consistency loss, as suggested in the shortcut, we select 1/4 to 1/8 of the data in each batch to calculate the multi-step consistency loss, and the remaining data is used to calculate the standard flow matching loss. We selected six groups. The first group selects 8 steps throughout the training process. The second group selects 4 steps throughout the training process. The third group uses 8 steps first and then 2 steps. The fourth group uses 4 steps first and then 2 steps. The fifth group randomly selects among



(a) Object classification (b) Meal packaging (c) Dynamic placement

Fig. 7: Evaluation in the real world.



(a) Peg-in-hole

(b) Wiping

Fig. 8: The experiments in the real world with multimodal tasks.

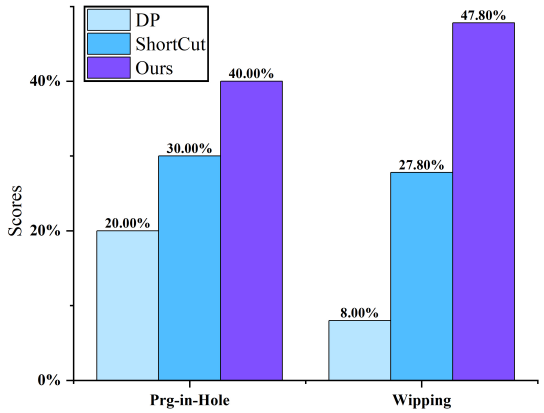


Fig. 9: Scores on real-world task with Multimodal.

8, 4, and 2. The last group uses 4 steps first, then 8 steps, and finally 2 steps. We also conducted relevant experiments on five tasks on Robotwin. The detailed experimental results are shown in the Table IV. We found that splitting into 4 steps is relatively reasonable, and appropriately mixing the number of steps is also effective.

To determine whether the proposed adaptive gradient adjustment method can effectively improve the algorithm's performance, we conducted ablation experiments based on five tasks on Robotwin. The detailed experimental results are shown in Table V. We found that after removing the gradient adjustment, the performance on some tasks dropped sharply. For example, on the block hammer beat task, the performance decline reached 65.2%, and there were also different degrees of reduction on the other four tasks. These experimental results indicate that the adaptive gradient adjustment algorithm proposed in this paper can well integrate the two losses and improve the algorithm's performance.

To further explore the mechanisms underlying the effectiveness of our method, we visualized the neural network's error landscape. Specifically, we applied the technique proposed in [31] to the block-hammer task, visualizing one-step prediction error as the metric; results are shown in Fig. 10. We compared the error landscapes with and without the AGA algorithm. From the 3D topography (i), incorporating AGA eliminates plateau regions in the error surface, yielding lower overall error values. The contour maps (ii) and (iii) further reveal that applying AGA results in sparser contours, indicating smoother transitions and broader coverage under equivalent error thresholds. These results confirm that the proposed AGA algorithm reduces the one-step error and improves the network's generalization capability

To examine how the initial value of c in Eq. 12 affects algorithm performance, we conducted additional ablation experiments. As before, five Robotwin tasks were used for ablation, and the results are presented in Table VI. The initial value shows minimal impact on most tasks but a noticeable effect on the block-hammer-beat task. Combined with previous results, this indicates that the task is more difficult to optimize and highly sensitive to gradient variations.

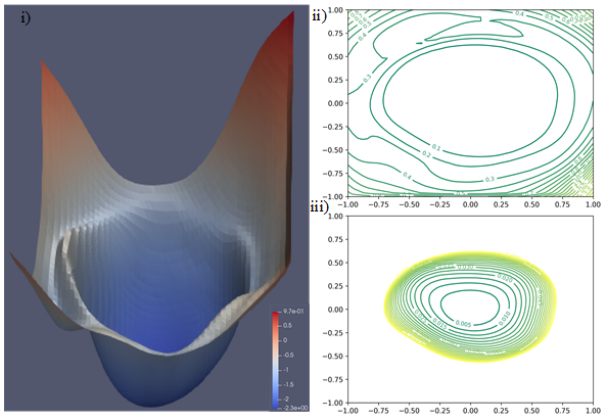
To examine how the NFE affects the algorithm's inference performance, we conducted ablation studies on five Robotwin tasks. As shown in Table VII, the proposed algorithm achieves

TABLE IV: The influence of computing consistency losses with different step numbers

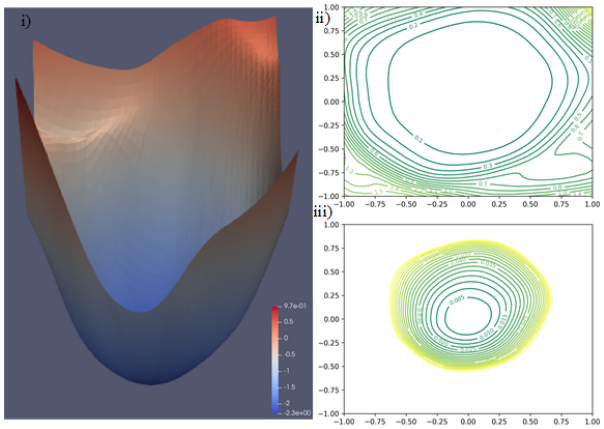
Task	8	4	8-2	4-2	random	4-8-2
bottle adjust	40.0±6.4	66.3±4.9	52.3±3.3	64.0±9.8	**67.3±5.0**	59.0±15.7
block hammer beat	59.0±5.3	66.3±1.3	62.0±9.2	74.0±7.8	63.3±3.4	**81.3±9**
dual bottles pick (easy)	90.0±3.5	**91.7±0.9**	86.0±2.9	89.3±2.5	89.7±4.1	89.3±0.5
diverse bottles pick	45.7±12.0	55.3±5.3	53.7±4.0	**58.3±3.3**	52.3±4.9	55.0±1.4
pick apple messy	7.7±3.4	**14.3±8.0**	9.0±4.2	11.0±6.5	6.3±1.7	13.7±9.2
average	48.5	58.8	52.6	59.2	55.8	**59.7**

TABLE V: The influence of AGA

Task	with	without
bottle adjust	59.0±15.7	**60.6±5.4**
block hammer beat	**81.3±0.9**	28.3±17.8
dual bottles pick (easy)	**89.3±0.5**	84.6±6.9
diverse bottles pick	**55.0±1.4**	48.6±4.0
pick apple messy	**13.7±9.2**	10.3±4.8



(a) With AGA



(b) Without AGA

Fig. 10: The Landscape of Neural Nets in bloack hammer.

TABLE VI: The influence of the initial threshold.

Task	1.0	0.5	0.01
bottle adjust	59.0±15.7	**63.7±5.9**	61.0±7.3
block hammer beat	**81.3±9**	73.7±8.8	64.6±6.5
dual bottles pick (easy)	**89.3±0.5**	88.0±2.9	85.7±7.6
diverse bottles pick	**55.0±1.4**	53.3±1.9	53.6±2.6
pick apple messy	13.7±9.2	**15.0±7.5**	14.3±5.7

strong inference performance with relatively few steps. Notably, the algorithm's performance varies across tasks. For example, in the bottle adjustment task—where the largest gap relative to 3DP is observed—performance improves as the number of inference steps increases, eventually matching 3DP at ten steps. Conversely, in the diverse bottle picking task, performance slightly decreases, though the decline remains modest. Overall, the algorithm exhibits a trend: performance improves initially but gradually declines as inference steps increase.

TABLE VII: The influence of the inference step.

Task	1	3	5	10
bottle adjust	59.0±15.7	64.0±11.3	67.7±11.5	**71.7±10.2**
block hammer beat	81.3±9	80.0±5.0	**82.3±1.7**	77.3±2.6
dual bottles pick (easy)	89.3±0.5	91.7±2.5	91.0±2.4	**92.0±2.2**
diverse bottles pick	**55.0±1.4**	51.7±3.6	54.0±2.8	51.3±0.4
pick apple messy	**13.7±9.2**	12.3±8.7	12.0±8.6	12.3±8.0
average	59.7	59.9	**61.4**	60.9

The experiments comprehensively address the four research questions presented in this section. For the first question, results demonstrate that the proposed method effectively handles diverse manipulation tasks. For the second question, three real-world experiments confirm the effectiveness of the proposed method. For the third question, real-world experiments on two rich-contact manipulation tasks indicate that the method generalizes well to multimodal fusion perception. Finally, ablation experiments verify that gradient treatment indeed enhances algorithm performance.

V. Conclusion

This study proposes a one-step flow-matching algorithm built upon multi-step integration and successfully applies it to robotic tasks, achieving competitive performance. To enhance training stability, an adaptive gradient allocation method is introduced, which treats the flow-matching and consistency losses as a specific form of multi-task learning. The effectiveness of the proposed method is further validated through extensive simulation and real-world experiments. In particular, results from rich-contact tasks integrating visual–tactile perception demonstrate the strong competitiveness of the proposed method. Finally, ablation experiments are conducted on each component to evaluate its individual contribution to overall performance. Future work will explore integrating vision–language large models to further enhance inference efficiency in related tasks.

APPENDIX A

Here we give the detailed derivation of the closed-form solution of our AGA. The goal of our method is:

$$c(\mathbf{g}\mathbf{u}_1^\top) = \mathbf{g}\mathbf{u}_2^\top \tag{17}$$

Given conditions: $\mathbf{g} = \alpha_1 g_1 + \alpha_2 g_2$, $\alpha_1 + \alpha_2 = 1$, $g_1 = Au_1$, $g_2 = Bu_2$, $\delta = u_1 \cdot u_2$. Substituting $\mathbf{g} = \alpha_1 g_1 + \alpha_2 g_2$ and $\delta = u_1 \cdot u_2$, we obtain:

$$c \cdot [\alpha_1 A + \alpha_2 B\delta] = \alpha_1 A\delta + \alpha_2 B \tag{18}$$

Substituting $\alpha_2 = 1 - \alpha_1$ yields:

$$c(\alpha_1 A + (1-\alpha_1)B\delta) = \alpha_1 A\delta + (1-\alpha_1)B \tag{19}$$

Then, after simplification, we get:

$$\alpha_1(cA - cB\delta) + cB\delta = \alpha_1(A\delta - B) + B \tag{20}$$

Finally, we can get

$$\alpha_1 = \frac{B(1-c\delta)}{(cA - A\delta) + B(1-c\delta)} \tag{21}$$

APPENDIX B

The constraint that the value of c should satisfy during the dynamic adjustment process can be derived based on $0 < \alpha_1 < 1$ and $\alpha_1 = \frac{B(1-c\delta)}{(cA-A\delta)+B(1-c\delta)}$. Thus, we obtain

$$0 < \frac{B(1-c\delta)}{A(c-\delta) + B(1-c\delta)} < 1 \tag{22}$$

Since the multi-step consistency loss depends on the flow-matching loss, we define c to be within the interval $[0, 1]$, always ensuring that the flow-matching loss dominates the direction of the composite gradient. Given the definition interval of c, the numerator is naturally positive. Based on $\frac{B(1-c\delta)}{(cA-A\delta)+B(1-c\delta)} < 1$, we can directly derive the first constraint:

$$\delta < c \tag{23}$$

Given that $B(1-c\delta) > 0$, it follows that:

$$c(A - B\delta) > (A\delta - B) \tag{24}$$

When $A > B\delta$, we have

$$c > \frac{(A\delta - B)}{(A - B\delta)} \tag{25}$$

When $A < B\delta$, we have

$$c < \frac{(A\delta - B)}{(A - B\delta)} \tag{26}$$

When $A = B\delta$, given that $-1 < \delta < 1$, in this case $0 < \delta < 1$ and $A < B$, thus $A\delta - B < 0$ holds. At this case, we have:

$$\frac{B(1-c\delta)}{B - A\delta} < 1 \tag{27}$$

Then we have :

$$\frac{A}{B} < c \tag{28}$$

In summary, we can derive all the constraints for c:

$$\begin{cases} c < \dfrac{A\delta - B}{A - B\delta} & \textbf{if} \quad \delta < c \quad \textbf{and} A < B\delta \\ \dfrac{A}{B} \quad < c & \textbf{if} \quad \delta < c \quad \textbf{and} A = B\delta \\ \dfrac{A\delta - B}{A - B\delta} \quad < c & \textbf{if} \quad \delta < c \quad \textbf{and} A > B\delta \end{cases} \tag{29}$$

REFERENCES

[1] Jiantao Yang, Tairen Sun, Long Cheng, and Zeng-Guang Hou. Spatial repetitive impedance learning control for robot-assisted rehabilitation. *IEEE/ASME Transactions on Mechatronics*, 28(3):1280–1290, 2023.

[2] Xiaofeng Xiong, Florentin Wörgötter, and Poramate Manoonpong. Adaptive and energy efficient walking in a hexapod robot under neuromechanical control and sensorimotor learning. *IEEE Transactions on Cybernetics*, 46(11):2521–2534, 2016.

[3] Fan Yang, Wenrui Chen, Haoran Lin, Sijie Wu, Xin Li, Zhiyong Li, and Yaonan Wang. Task-oriented tool manipulation with robotic dexterous hands: A knowledge graph approach from fingers to functionality. *IEEE Transactions on Cybernetics*, 55(1):395–408, 2025.

[4] Tianyu Fu, Yunfeng Bai, Cheng Li, Fengming Li, Chaoqun Wang, and Rui Song. Human-robot deformation manipulation skill transfer: Sequential fabric unfolding method for robots. *IEEE Robotics and Automation Letters*, 8(12):8454–8461, 2023.

[5] Ligang Jin, Yu Men, Fengming Li, Chaoqun Wang, Xincheng Tian, Yibin Li, and Rui Song. Ensemble transfer strategy based on domain difference for robot multiple peg-in-hole assembly. *IEEE Transactions on Industrial Electronics*, 71(10):12645–12654, 2024.

[6] Haoyu Zhang and Long Cheng. Boosting action-information via a variational bottleneck on unlabelled robot videos. *arXiv preprint arXiv:2508.08743*, 2025.

[7] Hao Sun, Yuanming Zhang, Huiyan Zhang, Xuan Qiu, and Imre J. Rudas. Learning distance constrained transformation for video tracking in car-following. *IEEE Transactions on Cybernetics*, 55(8):3747–3759, 2025.

[8] Zan Gao, Yibo Zhao, Hua Zhang, Da Chen, An-An Liu, and Shengyong Chen. A novel multiple-view adversarial learning network for unsupervised domain adaptation action recognition. *IEEE Transactions on Cybernetics*, 52(12):13197–13211, 2022.

[9] Zhengcai Cao, Junnian Li, Shibo Shao, Dong Zhang, and MengChu Zhou. Siamese adaptive network-based accurate and robust visual object tracking algorithm for quadrupedal robots. *IEEE Transactions on Cybernetics*, 55(3):1264–1276, 2025.

[10] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

[11] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. *arXiv preprint arXiv:2410.12557*, 2024.

[12] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[13] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *2nd Workshop on Dexterous Manipulation: Design, Perception and Control (RSS)*.

[14] Aaditya Prasad, Kevin Lin, Jimmy Wu, Linqi Zhou, and Jeannette Bohg. Consistency policy: Accelerated visuomotor policies via consistency distillation. *arXiv preprint arXiv:2405.07503*, 2024.

[15] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*.

[16] Max Braun, Noémie Jaquier, Leonel Rozo, and Tamim Asfour. Riemannian flow matching policy for robot motion learning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5144–5151. IEEE, 2024.

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

[18] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

[19] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

[20] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pages 87–103. Springer, 2024.

[21] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6613–6623, 2024.

[22] Bo Chen, Chengyue Gong, Xiaoyu Li, Yingyu Liang, Zhizhou Sha, Zhenmei Shi, Zhao Song, and Mingda Wan. High-order matching for one-step shortcut diffusion models. *arXiv preprint arXiv:2502.00688*, 2025.

[23] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *ArXiv*, abs/2505.13447, 2025.

[24] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5824–5836. Curran Associates, Inc., 2020.

[25] Shikun Liu, Edward Johns, and Andrew J. Davison. End-to-end multi-task learning with attention. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1871–1880, 2019.

[26] Liyang Liu, Yi Li, Zhanghui Kuang, Jing-Hao Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *International Conference on Learning Representations*, 2021.

[27] Yu Fang, Xuehe Zhang, Haoshu Cheng, Xizhe Zang, Rui Song, and Jie Zhao. Flow policy: Generalizable visuomotor policy learning via flow matching. *IEEE/ASME Transactions on Mechatronics*, pages 1–11, 2025.

[28] Guy Ohayon, Hila Manor, Tomer Michaeli, and Michael Elad. Compressed image generation with denoising diffusion codebook models, 2025.

[29] Yao Mu, Tianxing Chen, Shijia Peng, Zanxin Chen, Zeyu Gao, Yude Zou, Lunkai Lin, Zhiqiang Xie, and Ping Luo. Robotwin: Dual-arm robot benchmark with generative digital twins (early version). *arXiv preprint arXiv:2409.02920*, 2024.

[30] Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In *Conference on Robot Learning*, pages 1025–1037. PMLR, 2020.

[31] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.