

FAST VISUOMOTOR POLICY FOR ROBOTIC MANIPULATION

Anonymous authors
 Paper under double-blind review

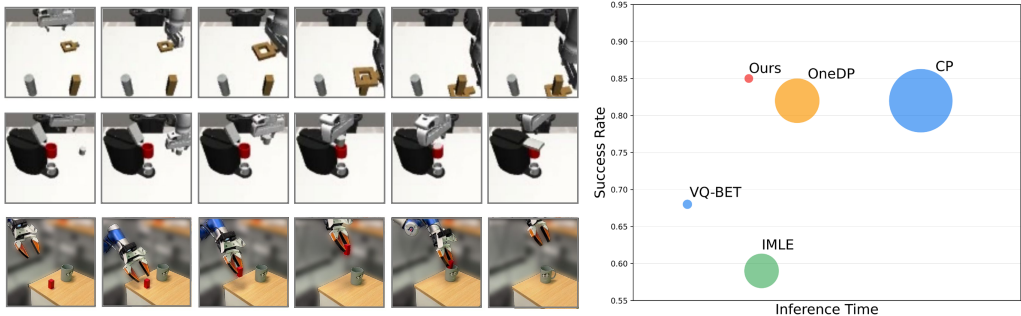


Figure 1: **Left:** Our method achieves strong results in three settings: single-task (top), multi-task (middle), and real-world (bottom). Each row displays rollouts from representative tasks within each scenario. **Right:** The bubble chart illustrates a representative comparison on PushT (Chi et al., 2023), showing that our method achieves state-of-the-art success rates while being faster in inference than the strongest baselines. Bubble sizes indicate model parameter counts, demonstrating that our approach delivers competitive performance with significantly smaller models.

ABSTRACT

We present a fast and effective policy framework for robotic manipulation, named **Energy Policy**, designed for high-frequency robotic tasks and resource-constrained systems. Unlike existing robotic policies, Energy Policy natively predicts multimodal actions in a single forward pass, enabling high-precision manipulation at high speed. The framework is built upon two core components. First, we adopt the energy score as the learning objective to facilitate multimodal action modeling. Second, we introduce an energy MLP to implement the proposed objective while keeping the architecture simple and efficient. We conduct comprehensive experiments in both simulated environments and real-world robotic tasks to evaluate the effectiveness of Energy Policy. The results show that Energy Policy matches or surpasses the performance of state-of-the-art manipulation methods while significantly reducing computational overhead. Notably, on the MimicGen benchmark, Energy Policy achieves superior performance with at a faster inference compared to existing approaches.

1 INTRODUCTION

Policy learning from demonstrations has emerged as a powerful paradigm for enabling robots to acquire complex skills. It is typically formulated as a supervised regression task, where observations are mapped to actions. To achieve high-precision action regression, incorporating generative models into policy learning has become a dominant approach across various robotic tasks.

Recent research has aimed to enhance policy learning by introducing various generative modeling techniques. Adopting Autoregressive Modeling (AM) from large language models (Kim et al., 2024; Brohan et al., 2022; 2023; Qu et al., 2025) has proven to be a powerful solution, owing to its scalability, flexibility, and mature exploration. However, like language models, AM uses discrete action tokens for action prediction, which can sacrifice fine-grained action details. Recently, there has

054 been considerable research into continuous action representations. While directly applying L1 or L2
055 regression (Kim et al., 2025; Su et al., 2025) to continuous action prediction offers a straightforward
056 way to improve action precision, it struggles with multimodal action distributions due to its uni-modal
057 modeling approach. Diffusion Modeling (DM) (Chi et al., 2023; Ze et al., 2024a; Wang et al., 2024b;
058 Team et al., 2024; Black et al., 2024; Intelligence et al., 2025) provides a promising alternative
059 by learning multimodal distributions through modeling the gradient of the action score function.
060 However, DM requires multiple denoising steps, making it computationally prohibitive for real-time
061 robotic tasks.

062 In this paper, we propose a novel and efficient approach to policy learning that natively predicts
063 multimodal continuous actions in a single forward pass. Specifically, during training, we utilize
064 energy score (Brier, 1950; Gneiting & Raftery, 2007b) as the learning objective to minimize the
065 distributional difference between the predicted actions and the ground truth. The energy score
066 provides a rigorous measure of whether predictions match the underlying distribution, making it a
067 natural choice for multimodal action modeling. To fully exploit this objective, we propose an energy
068 MLP, a dedicated module that explicitly parameterizes energy score modeling. This design is central
069 to our method, as it enhances representational expressiveness, allowing the energy score to serve
070 as an effective supervisory signal for complex multimodal distributions. During inference, we can
071 directly sample continuous actions from the model’s distribution prediction, avoiding the need for
072 multiple forward passes as in Diffusion Modeling. In addition, we incorporate parallel decoding,
073 which generates all actions simultaneously and enables efficient action chunking (Zhao et al., 2023).

074 We conduct extensive experiments to demonstrate the effectiveness of our proposed method. Across
075 a range of simulated robotic manipulation benchmarks, such as Robomimic (Mandlekar et al., 2021)
076 and MimicGen (Mandlekar et al., 2023), our method achieves high task success rates and fast
077 inference speeds. Notably, it outperforms CARP (Gong et al., 2024) across all benchmarks, with a
078 $2.3\times \sim 7\times$ faster inference speed, and further surpasses existing efficient policies on the PushT task.
079 We also evaluate our approach on real-world tasks under compute-constrained conditions. Compared
080 to baseline methods, our method exhibits a higher success rate and faster inference, underscoring its
081 suitability for real-time robotic applications.

082 In summary, our contributions are as follows: First, we present a novel approach to policy learning
083 that models multimodal continuous actions. Second, the proposed method offers faster inference
084 speeds, making it suitable for real-time robotic tasks. Third, extensive experiments validate the
085 effectiveness of our method in both simulated and real-world robotic manipulation tasks.

086 2 RELATED WORK

089 **Learning Robotic Manipulation from Demonstrations.** Imitation learning enables robots to learn
090 to perform tasks demonstrated by experts. Recently, there are various approaches to be developed
091 for policy learning with different task constraints and control modalities. Autoregressive Modeling
092 (AM) (Kim et al., 2024; Brohan et al., 2022; 2023; Qu et al., 2025) provides next-token prediction
093 paradigm and use discrete action representation for manipulation learning. RT2 (Brohan et al., 2023)
094 takes language instructions and visual observations as input, and outputs discrete action tokens in an
095 auto-regressive manner. For high precision manipulation, enormous works (Kim et al., 2025; Su et al.,
096 2025; Chi et al., 2023; Ze et al., 2024a; Wang et al., 2024b; Team et al., 2024; Black et al., 2024;
097 Intelligence et al., 2025) explore continuous action representations. (Kim et al., 2025; Su et al., 2025)
098 applies L1 or L2 objectives to learn to predict continuous action. However, these methods struggle
099 with multimodal action distributions due to their uni-modal nature. Diffusion Policy (Chi et al., 2023)
100 is proposed to handle multimodal action distributions by adopting a conditional denoising diffusion
101 process, which involves multiple denoising steps. Unlike existing works, our method employs energy
102 score to learn to predict continuous multimodal action.

103 **Fast Visuomotor Policy Learning.** In addition to manipulation precision, inference speed is
104 another critical aspect of robotic policy. For AM-based models, integrating the KV-Cache technique
105 can significantly enhance inference speed. Recent work such as Fast (Pertsch et al., 2025) introduces
106 compressed action tokens to further improve runtime, while CARP (Gong et al., 2024) employs
107 a next-scale autoregressive paradigm to shorten prediction horizons. Diffusion-based approaches
typically rely on action chunking (Zhao et al., 2023) to achieve higher action throughput, and can

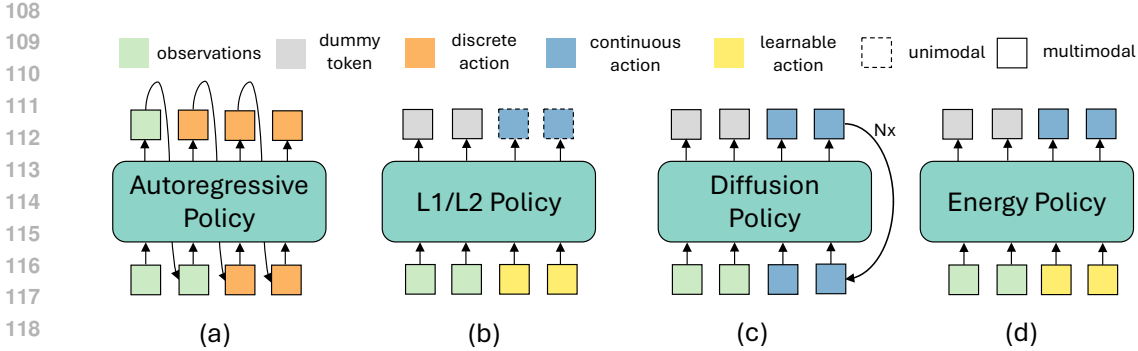


Figure 2: **Comparisons of existing Policies.** (a) Autoregressive policy predicts discrete tokens in an autoregressive manner. (b) L1/L2 policy predicts continuous actions but struggles with multimodal distribution modeling. (c) Diffusion policy generates multimodal continuous actions through multiple denoising steps. (d) Our Energy Policy produces multimodal continuous actions in a single forward pass.

leverage distillation techniques to reduce denoising steps (Prasad et al., 2024; Wang et al., 2024c), though often at the cost of action accuracy. Unlike these diffusion-based pipelines and their distilled variants, our approach natively predicts continuous actions in a single forward pass. This fundamental distinction eliminates the need for iterative refinement and avoids accuracy-compromising distillation, enabling our approach to achieve both high precision and fast inference simultaneously.

3 METHOD

In this section, we start by focusing on preliminaries, including problem formulation and existing works. Then we propose an energy-based learning objective to avoid these limitations. Finally, we demonstrate the details about network architecture to implement our method.

3.1 PRELIMINARIES

Problem Formulation. For a task \mathcal{T} , there are N expert demonstrations $\{\pi_i\}_{i=1}^N$. Each demonstration π_i consists of a sequence of state-action pairs $\{o_t, a_t\}_{t=1}^T$, where a_t denotes the action, o_t represents the observation, T is the action sequence length. We formulate robot policy learning as an action sequence prediction problem. The aim is to train a model to minimize the error in future actions conditioned on historical states. Specifically, policy learning minimize the imitation learning loss \mathcal{L}_{im} formulated as

$$\mathcal{L}_{im} = \mathbb{E}_{\pi_i \sim \mathcal{T}} \left[\sum_{t=0}^T \mathcal{L}(f_{\theta}(a_{t:t+H-1}|o_{t'<t}), a_{t:t+H-1}) \right] \quad (1)$$

where H is the prediction horizon, t and t' denote the current and previous time step, respectively. \mathcal{L} represents a supervised action prediction loss, and θ represents the learnable parameters of the policy network f_{θ} . Based on above problem formulation, the existing works (see Figure 2) mainly differ in how the \mathcal{L} and H are defined, discussed as follows.

Autoregressive Policy. By default, autoregressive-based policies predict action sequences in an autoregressive manner, with H set to 1. Additionally, due to the discrete nature of action tokens, cross-entropy loss is typically used as the default objective \mathcal{L} . However, using discrete action tokens often sacrifices fine-grained action details, making it challenging for robotic tasks that require high-precision control.

L1/L2 Policy. To circumvent the use of discrete action tokens, L1/L2 policies have been proposed. By using L1/L2 loss as the objective \mathcal{L} , these methods can predict continuous actions. This makes the learned policy well-suited for high-precision manipulation tasks. However, it struggles with multimodal action distributions due to its unimodal modeling characteristics.

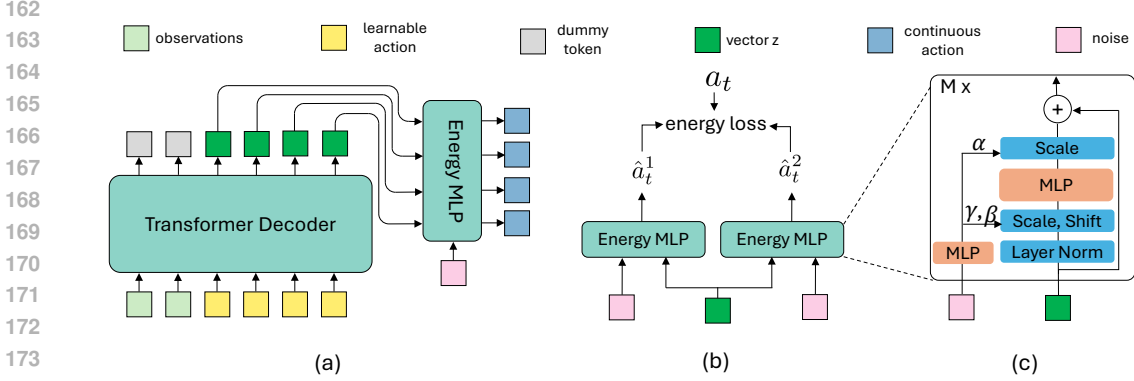


Figure 3: **Overview of Energy Policy.** (a) The architecture of the energy policy primarily consists of a transformer decoder and an energy MLP. The transformer decoder takes observations and learnable action tokens as input, producing a sequence of vectors $\{z_t\}_{t=1}^H$. The energy MLP then predicts the action sequence $\{\hat{a}_t\}_{t=1}^H$ by taking $\{z_t\}_{t=1}^H$ as input, conditioned on noise samples. (b) The energy loss is computed based on two sampled actions, \hat{a}_t^1 and \hat{a}_t^2 , along with the ground-truth action a_t . These two action samples are generated from the same z_t using different noise inputs. (c) The energy MLP is composed of several residual blocks, each incorporating adaLN for noise injection and modulation.

Diffusion Policy. In diffusion-based policies, action prediction is modeled as a denoising process, which makes it easier to handle multimodal action distributions. It uses denoising loss as the objective \mathcal{L} and predicts a sequence of actions with $H > 1$ simultaneously. However, diffusion-based policies suffer from the need for multiple denoising steps, making them less suitable for high-frequency robotic tasks.

3.2 ENERGY POLICY

To address the limitations of existing approaches, we propose an energy-based policy that uses energy scores as the learning objective. With energy scores, the model can learn multimodal action distributions during training. During inference, continuous actions can be sampled in a single forward pass. Additionally, we introduce a decoder-only transformer specifically designed for energy policy.

3.2.1 ENERGY LOSS

Consider two probability distributions p and q in \mathbb{R}^d . A scoring rule is a function $S(p, y)$ that assigns a score to the distribution p based on the observed data $y \sim q$ (Gneiting & Raftery, 2007a). The expected score $S(p, q)$ is defined as $S(p, q) := \mathbb{E}_{y \sim q}[S(p, y)]$. A scoring rule is called strictly proper if the expected score is minimized if and only if $p = q$.

A commonly used class of strictly proper scoring rules is the energy scores (Székely, 2003), defined as

$$S(p, y) = -\mathbb{E}[\|x_1 - x_2\|^\alpha] + 2\mathbb{E}[\|x - y\|^\alpha] \quad (2)$$

where $x_1, x_2, x \in \mathbb{R}^d$ are independent samples drawn from the model distribution p , and $\|\cdot\|$ denotes the Euclidean norm. By allowing for an index $\alpha \in (0, 2)$, $S(p, q)$ is minimized if and only if $p = q$, which implies that the model distribution is consistent with the data distribution.

For action prediction, the model takes an observation as input and outputs a prediction distribution p for the action a_t . To obtain an unbiased estimate of the energy score $S(p, q)$, two independent samples, \hat{a}_t^1 and \hat{a}_t^2 , are drawn from the model distribution p . The energy loss for action prediction is then defined as:

$$\mathcal{L}(p, a_t) = \|\hat{a}_t^1 - a_t\|^\alpha + \|\hat{a}_t^2 - a_t\|^\alpha - \|\hat{a}_t^1 - \hat{a}_t^2\|^\alpha \quad (3)$$

This loss objective incentivizes the model to generate samples close to the target action, while maintaining the diversity between independent samples. To implement the energy loss, we introduce an energy-based MLP within the transformer architecture.

3.2.2 TRANSFORMER WITH ENERGY MLP

As shown in Figure 3, our model takes observations and learnable action tokens as input. These input tokens are also combined with learnable position tokens. The decoder-only transformer then generates a sequence of vectors $\{z_t\}_{t=1}^H$, each corresponding to a learnable action token. For each vector z_t , the corresponding continuous ground-truth action is denoted as a_t .

Given ground-truth action a_t , we introduce a dedicated energy MLP specifically designed for the energy loss. This network takes z_t as input, together with two random noise samples drawn from a uniform distribution, and outputs two candidate actions \hat{a}_t^1 and \hat{a}_t^2 . The energy MLP consists of several residual blocks (He et al., 2016). Each block sequentially applies LayerNorm (LN) (Ba et al., 2016), a linear layer, SiLU (Elfwing et al., 2018), another linear layer, and a residual connection. To inject stochasticity, we adopt adaLN-Zero blocks (Peebles & Xie, 2022), which condition on noise inputs and perturb the predictions z_t through shift, scale, and gate operations. This design enables effective conditioning on noise, which not only introduces richer stochasticity but also enhances the model’s expressive power.

At inference time, this energy MLP takes z_t and a single noise sample to directly predict the corresponding continuous action, ensuring consistency between training and deployment while avoiding iterative refinement.

4 EVALUATION

We conduct a comprehensive evaluation of our method across a diverse set of robotic tasks in both simulated and real-world environments. These tasks include both single-task and multi-task settings, utilizing image-based and state-based observations. We compare our approach against several state-of-the-art baselines, including both diffusion-based and auto-regressive methods. The evaluation considers key metrics such as success rate, inference time, and model size. Our experimental study aims to address the following research questions:

- How does our method compare to state-of-the-art approaches in terms of inference speed and task success rate?
- How well does our method perform robotic tasks in real-world environments?
- Can our energy loss effectively learn the multimodal action distribution?

4.1 SIMULATION ENVIRONMENT

We evaluate our method across diverse settings, including single-object manipulation, long-horizon planning, high-precision control, and dual-arm coordination, using Robomimic (Mandlekar et al., 2021), Franka Kitchen (Gupta et al., 2019), MimicGen (Mandlekar et al., 2023), and PushT (Chi et al., 2023). Unless otherwise specified, the models are trained for 400 epochs with a batch size of 1024 and $\alpha = 1.0$, using an energy MLP head with a depth of 3 and a width of 512. At inference, the model predicts an action sequence of length 16, from which the first 8 actions are executed. Baseline models are trained and evaluated following their original protocols. Performance is reported as the average success rate over the best three checkpoints. Inference efficiency is measured on an NVIDIA RTX 4090 GPU by averaging the runtime for generating 8 executable actions, with each test repeated three times.

4.1.1 SINGLE OBJECT MANIPULATION ON ROBOMIMIC

Setup. Robomimic offers a diverse set of tasks; in this section, we focus on three representative single-object manipulation tasks: Lift, Can, and Square. For each task, we use 200 expert demonstrations collected via teleoperation in simulation and evaluate performance under two observation modalities: image-based (RGB inputs from eye-in-hand and third-person views) and state-based (low-dimensional privileged information). We compare against three baselines: Implicit Behavior Cloning (IBC) (Florence et al., 2021), Diffusion Policy (DP-C, DP-T) (Chi et al., 2023), and CARP (Gong et al., 2024).

Table 1: Comparison of policy performance and efficiency across state-based and image-based Robomimic (Mandlekar et al., 2021) tasks.

Policy	State-based					Image-based				
	Lift-ph	Can-ph	Square-ph	Params(M)	Speed(s)	Lift-ph	Can-ph	Square-ph	Params(M)	Speed(s)
IBC (Florence et al., 2021)	0.79	0.00	0.00	3.20	0.03	0.94	0.08	0.03	3.44	0.10
DP-C (Chi et al., 2023)	1.00	0.94	0.94	65.88	0.70	1.00	0.97	0.92	255.61	0.72
DP-T (Chi et al., 2023)	1.00	1.00	0.88	8.97	0.64	1.00	0.98	0.86	9.01	0.67
CARP (Gong et al., 2024)	1.00	1.00	0.98	0.65	0.07	1.00	0.98	0.88	7.58	0.11
Ours	1.00	1.00	0.97	0.73	0.01	1.00	0.98	0.95	11.51	0.03

Table 2: Performance and efficiency of different policies on Franka-Kitchen (Gupta et al., 2019) tasks.

Policy	p1	p2	p3	p4	Params(M)	Speed(s)
IBC (Florence et al., 2021)	0.99	0.87	0.61	0.24	3.28	0.05
DP-C (Chi et al., 2023)	1.00	1.00	1.00	0.96	66.94	0.91
DP-T (Chi et al., 2023)	1.00	0.99	0.98	0.96	80.42	0.84
CARP (Gong et al., 2024)	1.00	1.00	0.98	0.98	3.88	0.08
Ours	1.00	1.00	1.00	0.96	5.06	0.02

Result. As shown in Table 1, our model matches or surpasses the baselines in both state-based and image-based settings, while also delivering substantially faster inference. Specifically, our approach is $3.7\times \sim 7.0\times$ faster than the autoregressive baseline (CARP) and $22.3\times \sim 70.0\times$ faster than the diffusion-based baselines (DP-C and DP-T). Although achieving a low latency, IBC exhibits near-zero success rates on several tasks. In contrast, our method maintains high success rates comparable to the strongest baselines while eliminating the costly iterative sampling steps of diffusion. This demonstrates both efficiency and robustness across single-object manipulation tasks.

4.1.2 LONG HORIZON PLANNING ON FRANKA KITCHEN

Setup. To assess long-horizon, multi-task learning, we evaluate on the Franka Kitchen environment, which involves interaction with seven objects and 566 human demonstrations, each completing four tasks in arbitrary order. Only state-based inputs are used. We compare against three baselines: Implicit Behavior Cloning (IBC) (Florence et al., 2021), Diffusion Policy (DP-C, DP-T) (Chi et al., 2023), and CARP (Gong et al., 2024).

Result. As shown in Table 2, our model attains success rates that are comparable to or exceed those of the baselines across all tasks, while also maintaining the fastest inference speed. Specifically, our approach runs $4\times$ faster than the autoregressive baseline (CARP) and $42\times \sim 45\times$ faster than the diffusion-based baselines (DP-C and DP-T). These results highlight the efficiency and scalability of our design in complex, long-horizon environments.

4.1.3 MULTI-TASK LEARNING ON MIMICGEN

Setup. We evaluate our model on MimicGen (Mandlekar et al., 2023), a large-scale imitation learning benchmark, which extends Robomimic (Mandlekar et al., 2021) by providing 1K–10K demonstrations per task and broader initial state distributions. MimicGen comprises 12 MuJoCo (Todorov et al., 2012)-based robosuite tasks and 4 high-precision tasks from Isaac Gym Factory (Makoviy-chuk et al., 2021). Following prior work (Wang et al., 2024a; Gong et al., 2024), we evaluate on 8 robosuite tasks, each with 1K demonstrations. The baselines include two diffusion-based multitask models—Task-Conditioned Diffusion (TCD) (Liang et al., 2023) and Sparse Diffusion Policy (SDP) (Wang et al., 2024a), the latter employing a Transformer with a Mixture of Experts (MoE) (Shazeer et al., 2017)—as well as the autoregressive multitask variant of CARP (Gong et al., 2024). All models are conditioned on task labels to enable multitask learning.

Result. As shown in Table 3, our method achieves substantial improvements over the diffusion baseline, with about a 10% higher success rate and at least a $16.6\times$ faster inference. Moreover, it matches the success rate of the autoregressive baseline while delivering a $2.3\times$ faster inference. These results further support our first research question.

Table 3: Performance and efficiency comparison of different policies across 8 manipulation tasks in MimicGen (Chi et al., 2024).

Policy	Coffee	Hammer	Mug	Nut	Square	Stack	Stack 3	Threading	Avg.	Params (M)	Speed (s)
TCD (Liang et al., 2023)	0.77	0.92	0.53	0.44	0.63	0.95	0.62	0.56	0.68	156.11	1.33
SDP (Wang et al., 2024a)	0.82	1.00	0.62	0.54	0.82	0.96	0.80	0.70	0.78	159.85	1.53
CARP (Gong et al., 2024)	0.86	0.98	0.74	0.78	0.90	1.00	0.82	0.70	0.85	16.08	0.18
Ours	0.89	0.98	0.69	0.86	0.89	0.99	0.75	0.79	0.86	17.85	0.08

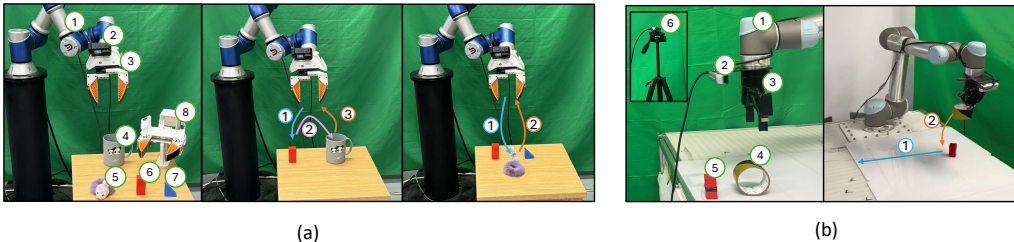


Figure 4: **Real-World Experiment Setup.** (a) Left: equipment and objects used in the static tasks, with each item labeled by a circled number in the upper-right corner. Middle: desired manipulation behavior for Task *Rabbit*. Right: desired manipulation behavior for Task *Cup*. (b) Left: equipment and objects used in the dynamic task, with each item labeled by a circled number in the upper-right corner. Right: desired manipulation behavior for Task *Catch*.

4.1.4 COMPARISON WITH OTHER EFFICIENT POLICIES

Setup. To further evaluate the performance of our method, we compare it against recent single-step robotic policies on more challenging tasks, including PushT, Square-mh, Square-ph, ToolHang-ph, Transport-mh, and Transport-ph, which challenge the model’s ability to learn dual-arm and high-precision manipulation. The baselines include One-step Diffusion Policy (OneDP) (Wang et al., 2024c), IMLE Policy (IMLE) (Rana et al., 2025), Consistency Policy (CP) (Prasad et al., 2024), and VQ-BeT (Lee et al., 2024). These methods are designed to improve the efficiency of robotic policy inference by eliminating or reducing reliance on multi-step sampling procedures.

Result. As summarized in Table 4, our Energy Policy consistently outperforms all baselines in average success rate across diverse tasks. It achieves an average success rate of 0.87, clearly surpassing OneDP, IMLE, CP, and VQ-BeT, while maintaining competitive inference speed. Although VQ-BeT attains the lowest latency, this comes at the cost of a substantial drop in task performance, underscoring the advantage of our approach in balancing efficiency and effectiveness.

Table 4: Comparison of different efficient policies across tasks. * For fair comparisons, we report the model parameter counts and inference speeds on the PushT task using an NVIDIA RTX 4090 GPU.

Policy	PushT	Square-mh	Square-ph	Toolhang-ph	Transport-mh	Transport-ph	Avg.	Params(M)*	Speed(ms)*
Ours	0.85	0.85	0.95	0.92	0.70	0.95	0.87	7.73	7.02
OneDP-S (Wang et al., 2024c)	0.82	0.86	0.93	0.85	0.69	0.91	0.84	251.51	9.33
IMLE (Rana et al., 2025)	0.59	-	0.82	0.81	-	0.90	-	75.75	7.63
CP (Prasad et al., 2024)	0.82	-	0.92	0.70	-	-	-	255.18	15.23
VQ-BeT (Lee et al., 2024)	0.68	-	-	-	-	-	-	4.37	4.09

4.2 REAL-WORLD ENVIRONMENTS

We evaluate our model on two robotic platforms to assess real-world manipulation performance. To systematically test different capabilities, we design three tasks: two static tasks (**Cup** and **Rabbit**) that assess precision and target selection under distraction, and one dynamic task (**Catch**) that evaluates real-time interaction. As the baseline, we use the CNN-based Diffusion Policy (Chi et al., 2023). Each model is evaluated from a single checkpoint, with success measured as the number of successful trials out of 20 per task. For speed evaluation, both models are deployed on the same machine with

Table 5: Comparison of policy performance and efficiency across real-world tasks.

Policy	Static Tasks				Dynamic Task			
	Cup	Rabbit	Params(M)	Speed(s)	Catch	Params(M)	Speed(s)	
DP-UMI (Chi et al., 2024)	16/20	19/20	85.47	0.34	DP-C (Chi et al., 2023)	8/20	64.87	0.10
Ours	17/20	20/20	68.65	0.10	Ours	13/20	11.50	0.02

an NVIDIA RTX 1080Ti GPU, and we report the average inference time for generating 8 executable action steps.

4.2.1 STATIC TASKS

Setup. As shown in Figure 4 (a, left), an Emergen CR3 robotic arm^① is equipped with a 3D-printed Universal Manipulation Interface (UMI) gripper^③ (Chi et al., 2024) and a wrist-mounted GoPro Hero 9 camera^② with a fisheye lens. we collect 150 human demonstration trajectories per task using the UMI gripper^③. As illustrated in Figure 4 (a, center and right), we design the following real-world tasks: **Cup.** The workspace contains a green cup^④ and a red block^⑥ on a tabletop. The robot must grasp the block and place it inside the cup. While the cup remains fixed, the block’s initial position is perturbed to introduce variability. This task evaluates precise pick-and-place under positional uncertainty. **Rabbit.** The workspace includes a soft rabbit toy^⑤, a red rectangular block^⑥, and a blue triangular block^⑦, which are randomly assigned to three predefined locations with small perturbations. The robot must identify and grasp the rabbit among distractors, testing the robustness to spatial variation and target ambiguity. As the baseline, we use the CNN-based Diffusion Policy (DP-UMI) (Chi et al., 2024).

Results. As shown in Table 5, our model achieves a slightly higher success rate than the baseline, while maintaining a $3.4\times$ inference speedup. Although the perturbations introduced during evaluation are relatively small, DP-UMI (Chi et al., 2024) occasionally fails, primarily due to slight gripping inaccuracies when the state deviates from the training distribution. In contrast, Energy Policy executes more reliably under the same conditions, underscoring its robustness in real-world robotic tasks.

4.2.2 DYNAMIC TASK

Setup. As shown in Figure 4 (b, left), a UR5 robotic arm^① is equipped with a Robotiq 2F-85 adaptive gripper^③ and a wrist-mounted Intel RealSense D435i camera^②. An additional RealSense D435i camera^⑥ mounted on a fixed stand provides a third-person view. We collect 100 teleoperated demonstrations using a 3D SpaceMouse. As illustrated in Figure 4 (b, right), we design the following task: **Catch.** The workspace contains a red block^⑤ attached to a string and a ring^④ held by the gripper. A human drags the block across the table at random speeds from right to left, and the robot must intercept it with the ring before it exits the camera’s field of view. This task challenges the policy’s inference speed and its ability to interact with fast-moving objects in real time. As the baseline, we use the CNN-based Diffusion Policy (DP-C) (Chi et al., 2023).

Results. As shown in Table 5, our model outperforms the CNN-based Diffusion Policy on the dynamic **Catch** task (13/20 vs. 8/20) with a $5\times$ faster inference speed (0.02s vs. 0.10s). The performance gap is more pronounced than in static tasks, since multi-step diffusion models struggle with latency. In contrast, our low-latency predictions enable reliable performance under time-sensitive conditions.

4.3 MODELING MULTI-MODAL BEHAVIOR

To evaluate the model’s ability to capture multimodal behavior, we design an initial state in the PushT environment where the task can be successfully completed by moving either left or right. We then sample 50 trajectories from the Energy Policy to examine its ability to model multi-modal behavior. The visualizations are shown in Figure 5, Energy Policy generates smooth trajectories covering both modes.



Figure 5: Rollout of the first 40 steps of the samples.

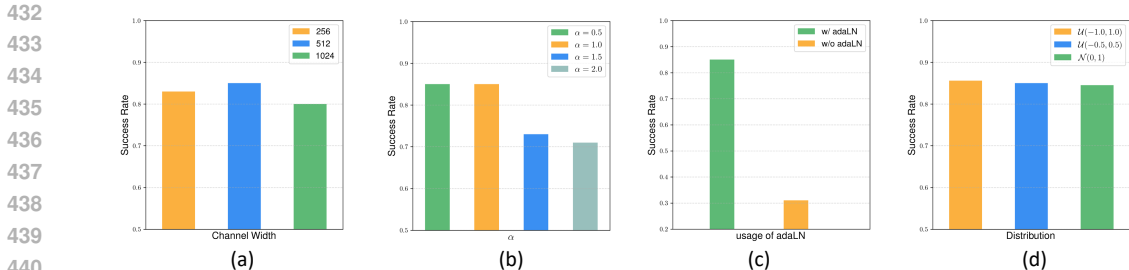


Figure 6: **Ablation Study Results.** (a) Success rate as a function of MLP channel width. (b) Success rate as a function of α . (c) Success rate with and without AdaLN. (d) Success rate under different noise injection distributions.

4.4 ABLATION STUDY

We conduct an ablation study to evaluate the contributions of the newly introduced components: the *Energy MLP* and the *Energy Loss*. All experiments are conducted on the *Square-mh* task from Robomimic (Mandlekar et al., 2021). Models are trained for 400 epochs with a batch size of 1024. For experiments involving the energy loss, we set the weighting coefficient $\alpha = 1.0$. Performance is assessed by averaging the success rates of the top three checkpoints. We investigate: (i) the impact of the Energy MLP channel width, (ii) the effect of the coefficient α , (iii) the role of adaLN, and (iv) the effect of different noise injection distributions.

Channel Size of Energy MLP. We investigate the impact of the Energy MLP channel width on model performance. Starting with a width equal to the token embedding size (256), we progressively increase the width to 512 and 1024. Performance improves from 0.83 to 0.85 when increasing the width to 512, but further increasing it to 1024 results in a decline to 0.80. We hypothesize that this degradation is due to overfitting. Based on these results, we adopt 512 as the default MLP width for all evaluations in the image-based simulation environment.

Choice of the coefficient α . In our main experiments, we set α to 1.0 empirically. To assess the sensitivity to this hyperparameter, we conduct an ablation study by varying its value. As shown in the Square-mh task, performance remains stable when reducing α from 1.0 to 0.5 (success rate 0.85), but degrades when increasing it to 1.5 (0.73) or 2.0 (0.71). Thus, we adopt $\alpha = 1.0$ as the default setting in all experiments.

Role of Adaptive Layer Normalization (adaLN). To validate its effectiveness, we compare our approach with a baseline that simply concatenates Gaussian noise with the output vector from the Transformer decoder, instead of applying adaptive layer normalization. We evaluate the success rate on the Square-mh task. Removing adaLN leads to a drastic performance drop from 0.85 to 0.31, highlighting that adaLN is crucial for effectively modeling noise conditioning.

Choice of Noise Distribution. By default, the noise injected into the Energy MLP is sampled from a uniform distribution in the range $[-0.5, 0.5]$. We also evaluated two alternative noise sources: a uniform distribution over $[-1.0, 1.0]$ and a Gaussian distribution $\mathcal{N}(0, 1)$. In all cases, the resulting success rates differ by less than 1%, indicating that our method is robust to variations in the noise injection distribution.

5 CONCLUSION

Energy Policy offers a novel approach to multimodal action modeling, making it well-suited for robotic tasks that demand high-precision manipulation. Moreover, thanks to its underlying mechanism, it can generate a sequence of continuous actions in a single forward pass, resulting in significantly faster inference. Our strong performance across a variety of robotic tasks demonstrates both the effectiveness and efficiency of the proposed method.

REFERENCES

- 486
487
488 Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint*
489 *arXiv:1607.06450*, 2016.
- 490
491 Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai,
492 Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0 : A vision-language-action flow model for
493 general robot control. *arXiv preprint arXiv:2410.24164*, 2024.
- 494
495 Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*,
496 78(1):1–3, 1950.
- 497
498 Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn,
499 Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics
transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.
- 500
501 Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski,
502 Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action
503 models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.
- 504
505 Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake,
506 and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The*
International Journal of Robotics Research, pp. 02783649241273668, 2023.
- 507
508 Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake,
509 and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-
510 wild robots. *ArXiv*, abs/2402.10329, 2024. URL [https://api.semanticscholar.org/](https://api.semanticscholar.org/CorpusID:267740127)
[CorpusID:267740127](https://api.semanticscholar.org/CorpusID:267740127).
- 511
512 Stefan Elfving, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network
513 function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018.
- 514
515 Peter R. Florence, Corey Lynch, Andy Zeng, Oscar Ramirez, Ayzaan Wahid, Laura Downs, Adrian S.
516 Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In
Conference on Robot Learning, 2021.
- 517
518 Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation.
519 *Journal of the American Statistical Association*, 102:359 – 378, 2007a. URL [https://api.](https://api.semanticscholar.org/CorpusID:1878582)
520 [semanticscholar.org/CorpusID:1878582](https://api.semanticscholar.org/CorpusID:1878582).
- 521
522 Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation.
523 *Journal of the American statistical Association*, 102(477):359–378, 2007b.
- 524
525 Zhefei Gong, Pengxiang Ding, Shangke Lyu, Siteng Huang, Mingyang Sun, Wei Zhao, Zhaoxin Fan,
526 and Donglin Wang. Carp: Visuomotor policy learning via coarse-to-fine autoregressive prediction.
arXiv preprint arXiv:2412.06782, 2024.
- 527
528 Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy
529 learning: Solving long-horizon tasks via imitation and reinforcement learning. *arXiv preprint*
530 *arXiv:1910.11956*, 2019.
- 531
532 Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image
533 recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*,
pp. 770–778, 2016.
- 534
535 Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess,
536 Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-
537 action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.
- 538
539 Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair,
Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source
vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

- 540 Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing
541 speed and success. *arXiv preprint arXiv:2502.19645*, 2025.
- 542
- 543 Seungjae Lee, Yibin Wang, Haritheja Etukuru, H Jin Kim, Nur Muhammad Mahi Shafiullah, and
544 Lerrel Pinto. Behavior generation with latent actions. *arXiv preprint arXiv:2403.03181*, 2024.
- 545
- 546 Zhixuan Liang, Yao Mu, Hengbo Ma, Masayoshi Tomizuka, Mingyu Ding, and Ping Luo. Skilldif-
547 fuser: Interpretable hierarchical planning via skill abstractions in diffusion-based task execution.
548 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16467–
549 16476, 2023. URL <https://api.semanticscholar.org/CorpusID:266361979>.
- 550 Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin,
551 David Hoeller, N. Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High
552 performance gpu-based physics simulation for robot learning. *ArXiv*, abs/2108.10470, 2021. URL
553 <https://api.semanticscholar.org/CorpusID:237277983>.
- 554 Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-
555 Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline
556 human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.
- 557
- 558 Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Ireteyayo Akinola, Yashraj Narang, Linxi Fan, Yuke
559 Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human
560 demonstrations. *arXiv preprint arXiv:2310.17596*, 2023.
- 561 William S Peebles and Saining Xie. Scalable diffusion models with transformers. 2023 ieee. In *CVF*
562 *International Conference on Computer Vision (ICCV)*, volume 4172, 2022.
- 563
- 564 Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees,
565 Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action
566 models. *arXiv preprint arXiv:2501.09747*, 2025.
- 567 Aaditya Prasad, Kevin Lin, Jimmy Wu, Linqi Zhou, and Jeannette Bohg. Consistency policy:
568 Accelerated visuomotor policies via consistency distillation. *arXiv preprint arXiv:2405.07503*,
569 2024.
- 570
- 571 Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu,
572 Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-
573 action model. *arXiv preprint arXiv:2501.15830*, 2025.
- 574 Krishan Rana, Robert Lee, David Pershouse, and Niko Suenderhauf. Imle policy: Fast and sample
575 efficient visuomotor policy learning via implicit maximum likelihood estimation. *arXiv preprint*
576 *arXiv:2502.12371*, 2025.
- 577
- 578 Noam M. Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E.
579 Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-
580 experts layer. *ArXiv*, abs/1701.06538, 2017. URL [https://api.semanticscholar.org/](https://api.semanticscholar.org/CorpusID:12462234)
581 [CorpusID:12462234](https://api.semanticscholar.org/CorpusID:12462234).
- 582 Yue Su, Xinyu Zhan, Hongjie Fang, Han Xue, Hao-Shu Fang, Yong-Lu Li, Cewu Lu, and Lixin Yang.
583 Dense policy: Bidirectional autoregressive learning of actions. *arXiv preprint arXiv:2503.13217*,
584 2025.
- 585
- 586 Gábor J. Székely. E-statistics: The energy of statistical samples. Technical Report 3.05, Bowling
587 Green State University, Department of Mathematics and Statistics, 2003.
- 588
- 589 Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep
590 Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot
591 policy. *arXiv preprint arXiv:2405.12213*, 2024.
- 592 Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control.
593 *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012.
URL <https://api.semanticscholar.org/CorpusID:5230692>.

594 Yixiao Wang, Yifei Zhang, Mingxiao Huo, Ran Tian, Xiang Zhang, Yichen Xie, Chenfeng Xu,
595 Pengliang Ji, Wei Zhan, Mingyu Ding, and Masayoshi Tomizuka. Sparse diffusion policy: A
596 sparse, reusable, and flexible policy for robot learning. In *Conference on Robot Learning*, 2024a.
597 URL <https://api.semanticscholar.org/CorpusID:270870600>.
598
599 Yixiao Wang, Yifei Zhang, Mingxiao Huo, Ran Tian, Xiang Zhang, Yichen Xie, Chenfeng Xu,
600 Pengliang Ji, Wei Zhan, Mingyu Ding, et al. Sparse diffusion policy: A sparse, reusable, and
601 flexible policy for robot learning. *arXiv preprint arXiv:2407.01531*, 2024b.
602
603 Zhendong Wang, Zhaoshuo Li, Ajay Mandlekar, Zhenjia Xu, Jiaojiao Fan, Yashraj Narang, Linxi
604 Fan, Yuke Zhu, Yogesh Balaji, Mingyuan Zhou, et al. One-step diffusion policy: Fast visuomotor
605 policies via diffusion distillation. *arXiv preprint arXiv:2410.21257*, 2024c.
606
607 Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan C. Julian, Karol Hausman, Chelsea Finn, and Sergey
608 Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning.
609 *arXiv preprint arXiv:1910.10897*, 2019.
610
611 Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion
612 policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv preprint*
613 *arXiv:2403.03954*, 2024a.
614
615 Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion
616 policy. *arXiv preprint arXiv:abs/2403.03954*, 2024b.
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647

Table 6: Evaluation on Meta-World.

Policy	Disassemble	Pick Place Wall	Shelf Place	Stick Pull	Stick Push	Avg.	Speed (ms)
Energy Policy (Ours)	0.95	0.85	0.30	0.70	0.75	0.71	7.72
DP3	0.95	0.90	0.25	0.60	0.80	0.70	73.33

A APPENDIX

A.1 MANIPULATION TASKS ON META-WORLD

Setup. We further extend our evaluation by adapting our method to the 3D Diffusion Policy Ze et al. (2024b) framework and testing on the Meta-World benchmark Yu et al. (2019). Following prior work Ze et al. (2024b), we evaluate five challenging tasks: *Disassemble*, *Pick Place Wall*, *Shelf Place*, *Stick Pull*, and *Stick Push*. This setting validates the ability of our model to handle complex, high-dimensional observation spaces.

Implementation. We adopt the same observation setup, encoder, and network architecture as 3D Diffusion Policy (DP3), with one key modification: the diffusion timestep embedding is removed, and a two-layer energy MLP (width 256) is added as the output head. Both models are trained for 1000 epochs with a batch size of 256.

Results. Table 6 summarizes the results. Our method achieves a higher average success rate than DP3 (0.71 vs. 0.70), while reducing inference latency by more than an order of magnitude (7.72 ms vs. 73.33 ms). These results highlight that our approach maintains competitive performance while delivering substantial efficiency gains in high-dimensional observation spaces settings.

A.2 EXPERIMENTS IMPLEMENTATION DETAILS

A.2.1 SINGLE-TASK SIMULATION (ROBOMIMIC, FRANKA KITCHEN, PUSHT)

We adopt the observation space configuration from Diffusion Policy (Chi et al., 2023). For image-based tasks, we use an observation horizon of 2, consisting of multiview RGB images and proprioceptive states. For state-based tasks, the observation horizon is 2 (Robomimic (Mandlekar et al., 2021)) or 4 (Franka Kitchen (Gupta et al., 2019)), using low-dimensional object state vectors as input. The action prediction horizon is set to 16, with only the first 8 actions executed during the evaluation. Our model, a decoder-only transformer with an energy-based MLP head, is trained with a batch size of 1024 for 400 epochs (image-based) or 600 epochs (state-based) using $\alpha = 1.0$. For the decoder-only Transformer, we use the same depth and token embedding dimension as DP-T. Baseline models are trained and evaluated following the original protocols.

A.2.2 MULTI-TASK SIMULATION (MIMICGEN)

We follow the observation space setup from Sparse Diffusion Policy (Wang et al., 2024a), using an observation horizon of 2 for image and robot pose input. Our model employs the same architecture as in the single-task evaluation, augmented with a task-class token and doubled network depth. Training is performed for 400 epochs with a batch size of 1024 and $\alpha = 1.0$. Baseline models are trained and evaluated according to their respective protocols. As in the single-task setting, the model predicts an action sequence of length 16, from which the first 8 actions are executed.

A.2.3 REAL-WORLD EXPERIMENTS

For all real-world tasks, we use an observation horizon of 2, incorporating camera RGB images and proprioceptive states.

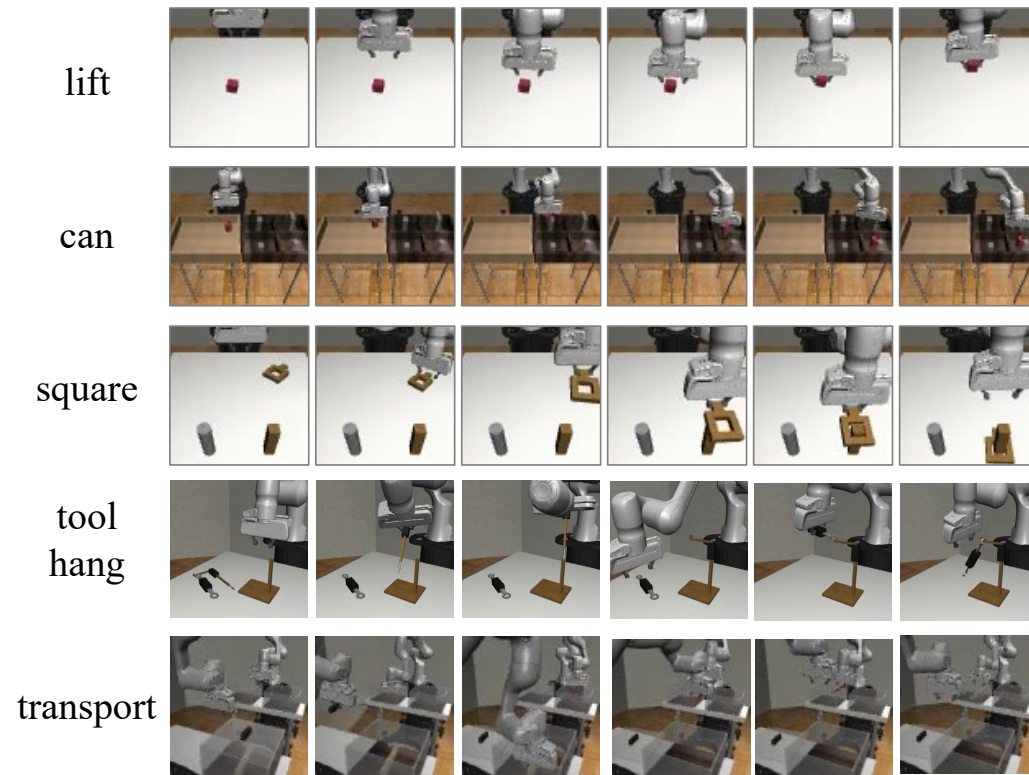
For the static tasks (*Cup* and *Rabbit*), our model retains the simulation architecture of a single task with modifications to address real-world complexity and noise: the transformer embedding size is tripled and the MLP channel width is doubled. Training is carried out for 150 epochs with a batch size of 64 and $\alpha = 1.0$. The baseline model follows its default architecture and training configuration. Both models predict an action sequence of length 16, with the first 8 actions executed.

702 For the dynamic task (*Catch*), our model uses the same architecture as in the single-task simulation
 703 setting without modification to the Transformer embedding size or MLP channel width. Training
 704 is carried out for 200 epochs with a batch size of 64 and $\alpha = 1.0$. The baseline model follows the
 705 configuration specified in its original implementation for real-world PushT.
 706

707 A.3 EXECUTION VISUALIZATION

708 We provide qualitative visualizations of successful rollouts across all benchmarks. In each figure, a
 709 row corresponds to a task rollout, with the leftmost frame showing the initial state and the rightmost
 710 frame showing the final state.
 711

712 A.3.1 ROBOMIMIC



741

742

743 Figure 7: Visualization of Robomimic experiments.

744

745

746

747

748

749

750

751

752

753

754

755

A.3.2 FRANKA KITCHEN

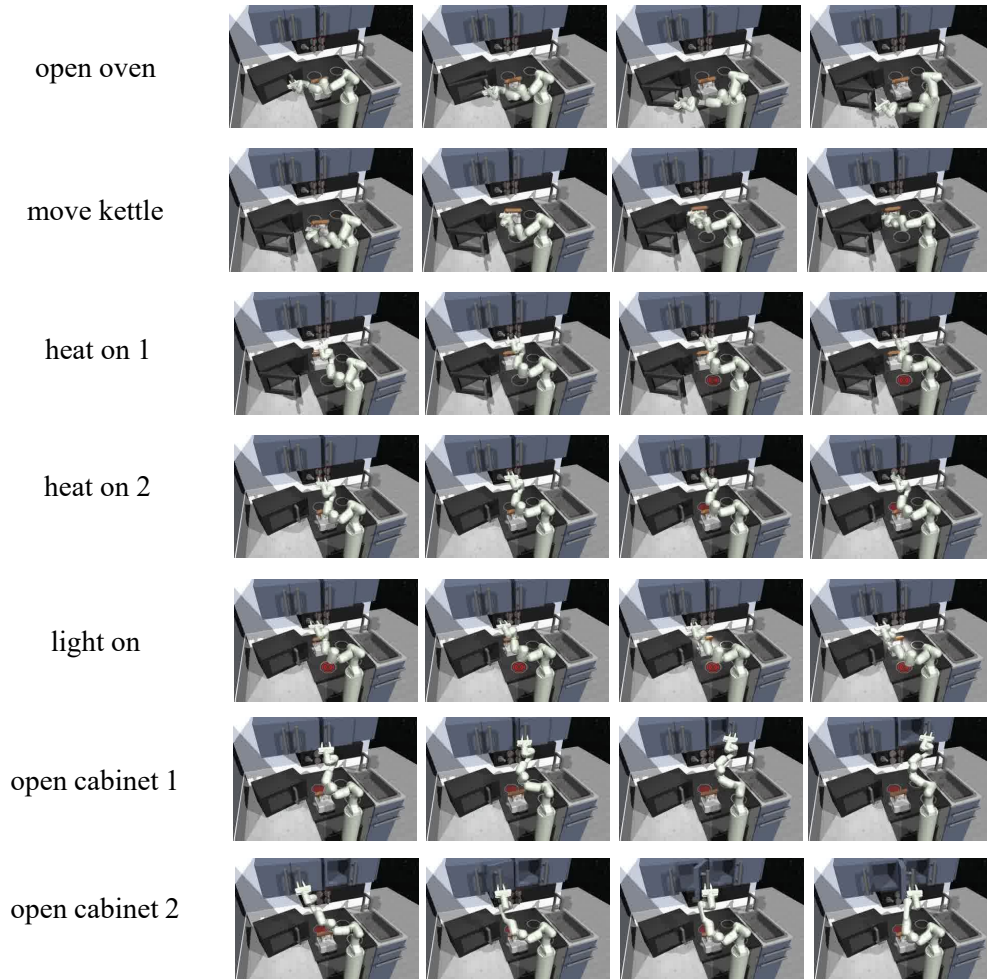


Figure 8: Visualization of Franka Kitchen subtasks.

756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809

A.3.3 MIMICGEN



Figure 9: Visualization of MimicGen multi-task experiments.

848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863

864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917

A.3.4 META-WORLD

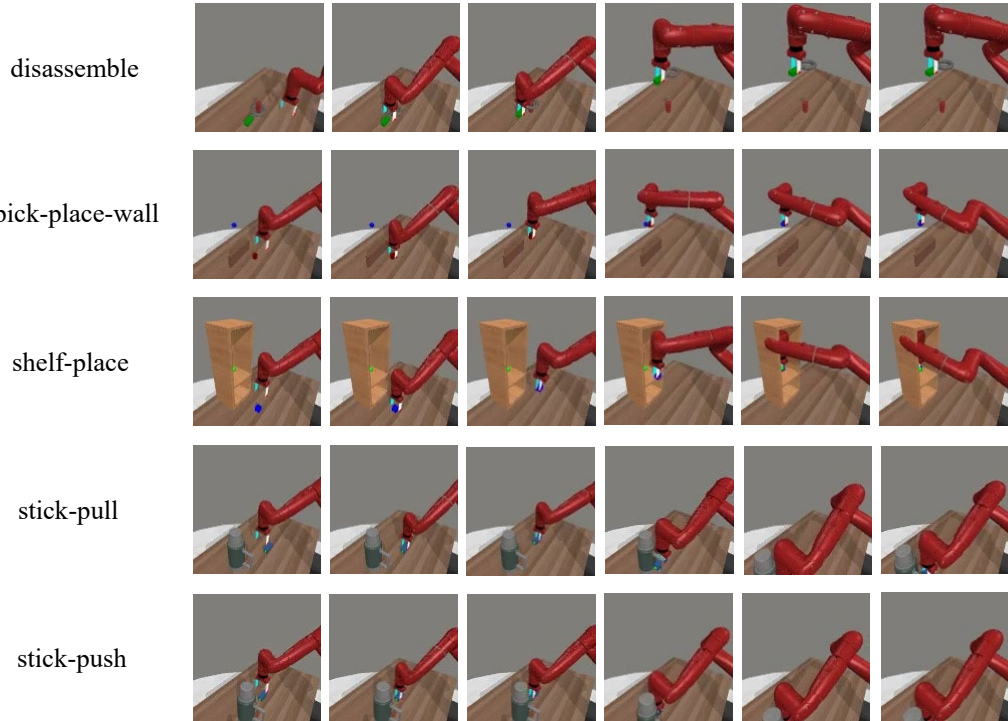


Figure 10: Visualization of Meta-World experiments.

918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971

A.3.5 REAL WORLD (STATIC)

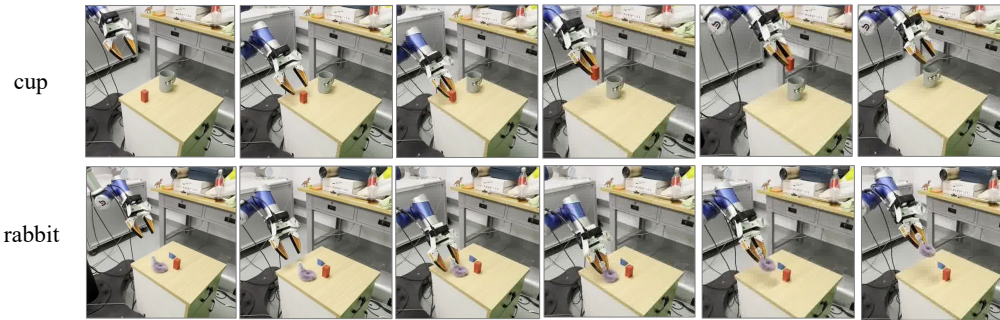


Figure 11: Visualization of real-world static task experiments.

A.3.6 REAL WORLD (DYNAMIC)

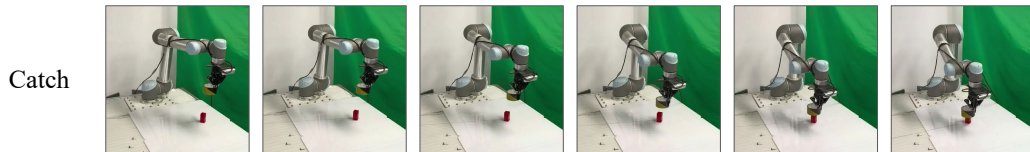


Figure 12: Visualization of real-world dynamic task experiments.

B LLM USAGE

Large Language Models (LLMs) were used for grammar checking and writing refinement.