CARP: Visuomotor Policy Learning via Coarse-to-Fine Autoregressive Prediction

Anonymous CVPR submission

Paper ID 3951

Abstract

*In robotic visuomotor policy learning, diffusion-based models have achieved significant success in improving the accuracy of action trajectory generation compared to traditional autoregressive models. However, they suffer from inefficiency due to multiple denoising steps and limited flexibility from complex constraints. In this paper, we introduce **C**oarse-to-fine **A**uto**R**egressive **P**olicy (**CARP**), a novel paradigm for visuomotor policy learning that redefines the autoregressive action generation process as a coarse-to-fine, next-scale approach. CARP decouples action generation into two stages: first, an action autoencoder learns multi-scale representations of the entire action sequence; then, a GPT-style transformer refines the sequence through a coarse-to-fine autoregressive process. This straightforward and intuitive approach produces highly accurate and smooth actions, matching or even surpassing the performance of diffusion-based policies while maintaining efficiency on par with autoregressive policies. We conduct extensive experiments in both simulated and real-world environments, demonstrating that CARP achieves competitive success rates, with up to a 5% improvement, and delivers **10×** faster inference compared to state-of-the-art policies, establishing a high-performance and efficient paradigm for action generation in robotic tasks.*

1. Introduction

Policy learning from demonstrations, formulated as the supervised regression task of mapping observations to actions, has proven highly effective across various robotic tasks, even in its simplest form. Replacing the policies with generative models, particularly those stemming from the vision community, has opened new avenues for improving performance, enabling robots to achieve the high precision required for complex tasks.

Existing approaches have explored different generative modeling techniques to address challenges in visuomotor policy learning. Autoregressive Modeling (AM) [14, 48, 66] provides a straightforward and efficient out-of-the-box solution, benefitting from its scalability, flexibility, and mature exploration at lower computation requirements. However, AM's next-token prediction paradigm often fails to capture long-range dependencies, global structure, and temporal coherence [25], leading to poor performance, which is essential for many robotic tasks. Recently, Diffusion Modeling (DM) [13, 57] has emerged as a promising alternative, bridging the precision gap in AM by modeling the gradient of the action score function to learn multimodal distributions. Nevertheless, DM requires multiple steps of sequential denoising, making it computationally prohibitive for robotic tasks, especially for robotic tasks requiring efficient real-time inference in on-board compute-constrained

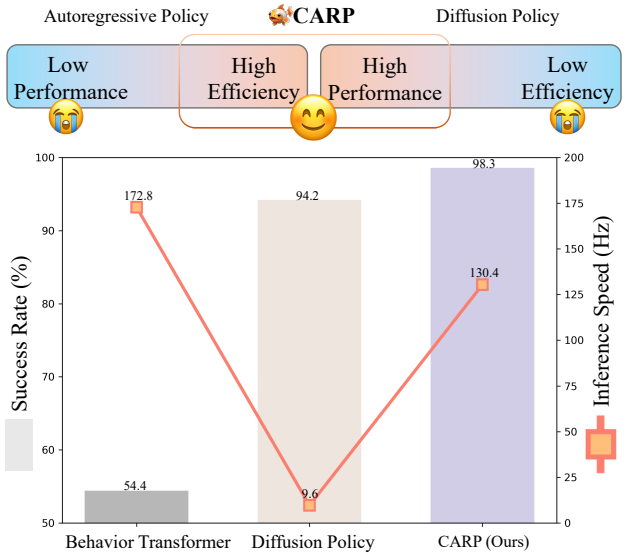
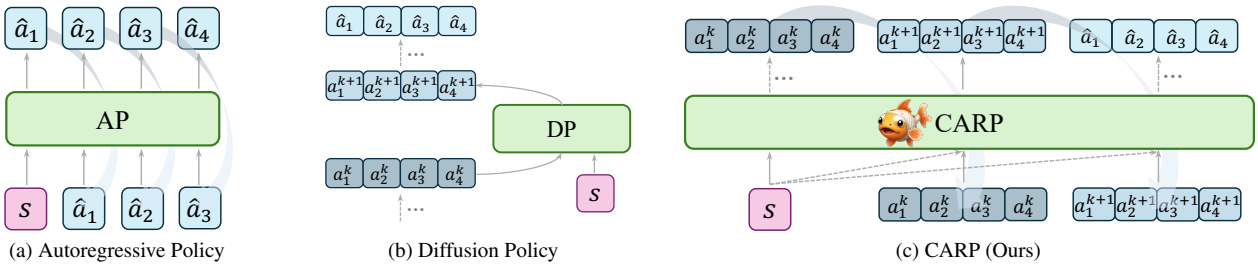


Figure 1. **Policy Comparison.** The representative performance among Behavior Transformer [48] served as an autoregressive policy, Diffusion Policy [13], and our approach in the state-based Robomimic square task experiment. CARP achieves a superior balance of performance and efficiency.



(a) Autoregressive Policy (b) Diffusion Policy (c) CARP (Ours)

Figure 2. **Structure of Current Policies.** $\hat{a}$ is the predicted action, a^k denotes the refining actions at step k, s is the historical condition. a) Autoregressive Policy predicts the action step-by-step in the next-token paradigm. b) Diffusion Policy models the noise process used to refine the action sequence. c) CARP refines action sequence predictions autoregressively from coarse to fine granularity.

environments. Additionally, DDPM [20]'s rigid generative process lacks adaptability for tasks with long-term dependencies, often leading to cumulative errors and reduced robustness over extended time spans.

Both AM and DM have their respective advantages and limitations, which are often orthogonal and difficult to balance in practical applications. In this work, we aim to resolve this trade-off by introducing a novel generative paradigm for robot visuomotor policy learning that predicts entire action sequences from a *coarse-to-fine* granularity in a *next-scale* prediction framework. This approach allows our model to achieve performance levels comparable to DM while achieving AM's efficiency and flexibility.

Our primary contribution is the introduction of a **Coarse-to-fine AutoRegressive Policy** (**CARP**), a hybrid framework that combines AM's efficiency with DM's high performance to meet the demands of real-world robotic manipulation. Specifically, our contributions are as follows:

- **Multi-scale action tokenization:** We propose a multi-scale tokenization method for action sequences that captures the global structure and maintains temporal locality, effectively addressing AM's myopic limitations.
- **Coarse-to-fine autoregressive prediction:** This mechanism refines action sequences in the latent space using Cross-Entropy loss with relaxed Markovian assumptions during iterations, achieving DM-like performance with high efficiency.
- **Comprehensive sim & real experiments:** Extensive experiments demonstrate CARP's effectiveness in both simulated and real-world robotic manipulation tasks.

In summary, we present CARP, a novel visuomotor policy framework that synergizes the strengths of AM and DM, offering high performance, efficiency, and flexibility. We will release the code and training details publicly to support reproducibility.

2. Background

We start by introducing the background, focusing on three key areas: problem formulation, conventional autoregressive policies, and diffusion-based policies.

2.1. Problem Formulation

Problem formulation will consider a task $\mathcal{T}$, where there are N expert demonstrations $\{\tau_i\}_{i=1}^N$. Each demonstration τ_i is a sequence of state-action pairs. We formulate robot imitation learning as an action sequence prediction problem [13, 43, 57], training a model to minimize the error in future actions conditioned on historical states. Specifically, imitation learning minimize the behavior cloning loss $\mathcal{L}_{bc}$ formulated as

$$\mathcal{L}_{bc} = \mathbb{E}_{s,a\sim\mathcal{T}}\left[\sum_{t=0}^{T}\mathcal{L}\left(\pi_{\boldsymbol{\theta}}(a_H|s_o), a_H\right)\right], \quad (1)$$

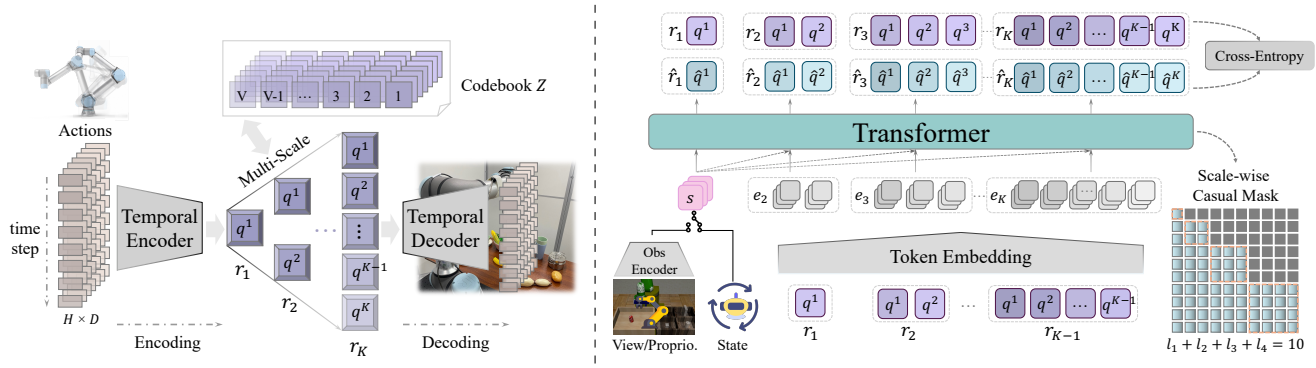
where a represents the action, s denotes the state, t is the current time step, H is the prediction horizon, and o is the historical horizon. For notational simplicity, we denote the action sequence $a_{t:t+H-1}$ as a_H and the state sequence $s_{t-o+1:t}$ as s_o. Here, $\mathcal{L}$ represents a supervised action prediction loss, such as mean squared error or negative log-likelihood, T is the total length of the demonstration, and θ represents the learnable parameters of the policy network $\pi_{\boldsymbol{\theta}}$.

2.2. Autoregressive Policy

The autoregressive policy is proposed naturally to utilize the significant efficiency and flexibility of autoregressive models, like [48, 66]. The next-token autoregressive posits the probability of observing the current action a_t depends on its prefix predictions $(a_1, a_2, ..., a_{t-1})$, which allows for the factorization of the likelihood of sequence with length H as

$$p\left(a_t, a_{t+1}, ..., a_{t+H-1}\right) = \prod_{k=t}^{t+H-1} p\left(a_k|a_{t:k-1}, s_o\right). \quad (2)$$

However, it introduces several issues to obstacle its performance. The linear, step-by-step unidirectional dependency of action prediction may overlook the global structure [25], making it challenging to capture long-range dependencies



(a) Multi-scale action tokenization

(b) Coarse-to-fine autoregressive prediction

Figure 3. **Overview of the Two Stages of CARP.** a) A multi-scale action autoencoder extracts token maps $r_1, r_2, \ldots, r_K$ to represent the action sequence at different scales, trained using the standard VQVAE loss. b) The autoregressive prediction is reformulated as a *coarse-to-fine*, *next-scale* paradigm. The sequence is progressively refined from coarse token map r_1 to finer granularity token map r_K, where each r_k contains l_k tokens. An attention mask ensures that each r_k attends only to the preceding $r_{1:k-1}$ during training. A standard Cross-Entropy loss is used for training. During inference, the final token map r_K is decoded into continuous actions for execution.

and holistic coherence or temporal locality [22] in complex scenes or sequences, which restricts their generalizability in tasks requiring bidirectional reasoning, as shown in Fig. 2a. For example, it cannot predict the former actions given the near-terminal state.

2.3. Diffusion Policy

Diffusion policy [13] utilizes Denoising Diffusion Probabilistic Models [20] to approximate the conditional distribution $p(a_H|s_o)$, through modeling the noise during the denoising process from Gaussian noise to noise-free output as

$$a_H^{k+1} = \alpha(a_H^k - \gamma\epsilon_\theta(s_o, a_H^k, k) + \mathcal{N}), \tag{3}$$

where ϵ_θ is the learnable noise network, k is the current denosing step, $\mathcal{N}$ is the Gaussian noise, α and γ are the hyper-parameters.

Diffusion policies show impressive performance in robotic manipulation tasks due to their action generation which gradually refines from random samples which we can abstract as a *coarse-to-fine* process. This kind of process alike human movement or natural thinking flows in line with human intuition. However, they suffer from poor convergence due to their design. To address this, multiple denoising steps are required, constrained by the Markovian assumption, where a^{k+1} depends only on the previous step a^k (as shown in Fig. 2b), which leads to significant runtime inefficiency. Moreover, they lack the ability to leverage generation context effectively, hindering scalability to complex scenes [18].

3. Method

To address the limitations of existing methods, we propose **C**oarse-to-fine **A**uto**R**egressive **P**olicy (**CARP**), a novel visuomotor policy framework that combines the high performance of recent diffusion-based policies with the sample efficiency and flexibility of traditional autoregressive policies.

CARP achieves these advantages through a redesigned autoregressive modeling strategy. Specifically, we shift from conventional *next-token* prediction to a *coarse-to-fine*, *next-scale* prediction approach, using *multi-scale* action representations. CARP's training is divided into two stages: *multi-scale* action tokenization and *coarse-to-fine* autoregressive prediction, as shown in Fig. 3.

In this section, we first explain the construction of *multi-scale* action token maps, followed by the *coarse-to-fine* autoregressive prediction approach for action generation. Key implementation details are provided at the end.

3.1. Multi-scale Action Tokenization

Instead of focusing on individual action steps, we extract representations at multiple scales across the entire action sequence. We propose a novel *multi-scale* action quantization autoencoder that encodes a sequence of actions into K discrete token maps, $R = (r_1, r_2, \ldots, r_K)$, which are used for both training and inference. Our approach builds on the VQVAE architecture [56], incorporating a modified *multi-scale* quantization layer [54] to enable hierarchical encoding.

Encoder and Decoder. As illustrated in Fig. 3a, the actions are first organized into an action sequence $A^{H\times D}$, where H denotes the prediction horizon and D represents the dimensionality of the action a. Given that each dimension in the action space is orthogonal to the others, and that there exists a natural temporal dependency within an action sequence, we employ a 1D temporal convolutional network (1D-CNN) along the time dimension, as used in [22], for

both the encoder $\mathcal{E}(\cdot)$ and decoder $\mathcal{D}(\cdot)$. Let $F = \mathcal{E}(A) \in \mathbb{R}^{L\times C}$, where L denotes the compressed length of the temporal dimension with $L \leq H$, and C represents the dimensionality of the feature map F.

Quantization. We introduce a quantizer with a learnable codebook $Z \in \mathbb{R}^{V\times C}$, containing V code vectors, each of dimension C. The quantization process (lines 4-11 in Algorithm 1) generates the action sequence representation iteratively across multiple scales. At scale k, it produces action token map r_k to represent the sequence, which consists of l_k tokens q. In our implementation, we set $l_k = k$. The function $\text{Lookup}(Z, v)$ retrieves the v-th code vector from Z. The quantization function is defined by $r = \mathcal{Q}(F)$ as following:

$$q = \underset{v\in[V]}{\arg\min}\, \text{dist}\left(\text{Lookup}(Z, v), f\right) \in [V], \tag{4}$$

where token q represents the nearest vector in Z for a given feature vector $f \in \mathbb{R}^{1\times C}$ in the feature map F, based on a distance function $\text{dist}(\cdot)$.

Specifically, we adopt a residual-style design [27, 54] for feature maps F and $\hat{F}$, as shown in lines 4-11 of Algorithm 1. This design ensures that each finer-scale representation r_k depends only on its coarser-scale predecessors $(r_1, r_2, \ldots, r_{k-1})$, facilitating a *multi-scale* representation. A shared codebook Z is used across all scales, ensuring that *multi-scale* token maps $R = (r_1, r_2, \ldots, r_K)$, where K is the number of scales, are drawn from a consistent vocabulary $[V]$. To preserve information during upsampling, we employ K additional 1D convolutional layers $\{\phi_k\}_{k=1}^{K}$ [54], as illustrated in lines 9-10 of Algorithm 1.

Loss. The final approximation $\hat{F}$ of the original feature map F is combined residually with the *multi-scale* representation derived from the codebook Z, based on each r. The reconstructed action sequence is then obtained as $\hat{A} = \mathcal{D}(\hat{F})$. To train the quantized autoencoder, a typical VQVAE loss $\mathcal{L}$ is minimized as following

$$\mathcal{L} = \underbrace{\|A - \hat{A}\|_2}_{L_{\text{recon}}} + \underbrace{\|\text{sg}(F) - \hat{F}\|_2}_{L_{\text{quant}}} + \underbrace{\|F - \text{sg}(\hat{F})\|_2}_{L_{\text{commit}}}, \tag{5}$$

where L_{recon} minimizes the difference between the original action sequence A and the reconstruction $\hat{A}$. $\text{sg}(\cdot)$ denotes stop-gradient. L_{quant} aligns the quantized feature map $\hat{F}$ with the original F, and L_{commit} encourages the encoder to commit to codebook entries, preventing codebook collapse. For L_{quant} and L_{commit}, we calculate every residual calculating moment of each scale r_k as in Algorithm 1. After training, the autoencoder $\{\mathcal{E}, \mathcal{Q}, \mathcal{D}\}$ tokenizes actions for subsequent *coarse-to-fine* autoregressive modeling.

Discussion. The tokenization strategy described above allows the *multi-scale* tokens R, extracted from the action sequence $A \in \mathbb{R}^{H\times D}$ via temporal 1D convolutions, to inherently preserve temporal locality [22]. Additionally, the hierarchical extraction captures the global structure, enabling the model to treat the action sequence as a unified entity.

Unlike traditional autoregressive policies that predict each action token independently (see Eq. (2) and Fig. 2a), CARP leverages dual capabilities to capture both local temporal dependencies and global structure across the entire action sequence. This approach produces smoother transitions and more stable action sequences. Through *multi-scale* encoding, CARP overcomes the short-sighted limitations of conventional autoregressive models, yielding more robust, coherent, and precise behaviors over extended time horizons.

Algorithm 1 Multi-scale Action VQVAE

1: **Inputs:** Action sequence A
2: **Hyperparameters:** Number of scales K, length of each scale $(l_k)_{k=1}^{K}$, length of feature map's temporal dimension L
3: **Initialize:** $F \leftarrow \mathcal{E}(A)$, $\hat{F} \leftarrow 0$, $R \leftarrow []$
4: **for** $k = 1$ **to** K **do**
5: $\quad r_k \leftarrow \mathcal{Q}(\text{Interpolate}(F, l_k))$
6: $\quad R \leftarrow R \cup \{r_k\}$
7: $\quad z_k \leftarrow \text{Lookup}(Z, r_k)$
8: $\quad z_k \leftarrow \text{Interpolate}(z_k, L)$
9: $\quad F \leftarrow F - \phi_k(z_k)$
10: $\quad \hat{F} \leftarrow \hat{F} + \phi_k(z_k)$
11: **end for**
12: $\hat{A} \leftarrow \mathcal{D}(\hat{F})$
13: **Return:** Multi-scale action tokens R, reconstructed action $\hat{A}$

3.2. Coarse-to-fine Autoregressive Prediction

Using *multi-scale* action sequence representations, we shift from traditional *next-token* prediction to a *next-scale* prediction approach, progressing from coarse to fine granularity.

Prediction. As we receive the *multi-scale* representation tokens $(r_1, r_2, ..., r_K)$, each r represents the same action sequence in different scales. The autoregressive likelihood can be formulated as

$$p(r_1, r_2, \ldots, r_K) = \prod_{k=1}^{K} p(r_k \mid r_1, r_2, \ldots, r_{k-1}; s_o), \tag{6}$$

where each autoregressive unit $r_k \in [V]^{l_k}$ is the token map at scale k containing l_k tokens q. The predix sequence $(r_1, r_2, \ldots, r_{k-1})$ is served as the condition for r_k, accompanying with the historical state sequence s_o. This kind of *next-scale* prediction methodology is what we define as *coarse-to-fine* autoregressive prediction.

Due to the residual-style quantization, during autoregressive prediction, we first embed the previous scale token map r_{k-1} and then reconstruct the next scale feature map e_k, as shown in Fig. 3b. This feature map e_k is then used as input

for predicting the next scale token map r_k. The final token map r_K is decoded by the *multi-scale* autoencoder into continuous actions for execution.

Loss. During the k-th autoregressive step, all distributions over the l_k tokens in r_k will be generated in parallel, with the *coarse-to-fine* dependency ensured by a block-wise causal attention mask [54]. To optimize the autoregressive model, we utilize the standard Cross-Entropy loss to capture the difference between the predicted token map $\hat{r}$ and the token map r from the ground truth action sequence:

$$\mathcal{L}_{\text{Cross-Entropy}} = \sum_{k=1}^{K} \sum_{i=1}^{l_k} \left[-\sum_{v=1}^{V} r_k^{i,v} \log \hat{r}_k^{i,v} \right], \quad (7)$$

where l_k is the length of each scale, V is the action vocabulary size, and K is the number of scales.

Discussion. CARP models the entire trajectory holistically, progressively refining actions from high-level intentions to fine-grained details (see Fig. 2c). This approach aligns more closely with natural human behavior, where movement is guided by overarching intentions rather than step-by-step planning.

Instead of the commonly used MSE loss [6, 55, 66], CARP employs Cross-Entropy loss (Eq. (7)) in behavior cloning, as MSE tends to enforce a unimodal distribution that can be detrimental to manipulation tasks [48]. CARP's iterative refinement process resembles the denoising steps in diffusion models to achieve high accuracy.

Furthermore, our approach models actions directly rather than modeling noise, enabling faster convergence in low-dimensional manifolds [32, 61]. By operating in the latent space with action sequence tokens, CARP mitigates trajectory anomalies that can disrupt prediction, unlike methods that work on raw actions [13]. This latent-space representation allows CARP to focus on the essential components of actions, resulting in smoother and more efficient predictions. In contrast, traditional diffusion models rely on Markovian processes, which compress all information into progressively noisier inputs from previous levels, often hindering efficient learning and requiring more inference steps [18] (see Eq. (3) and Fig. 2b). CARP relaxes this constraint by allowing each scale r_k to depend on all prior scales $r_{1:k-1}$ instead of the only previous scale r_{k-1}. This structure enables CARP to generate high-quality trajectories with significantly fewer steps.

3.3. Implementation Details

Our primary focus is on the CARP algorithm, and we adopt a simple model architecture design.

Tokenization. Due to the disentangling of the action prediction, imprecise *coarse-to-fine* actions representation can decrease the upper limit of model performance. Considering the discontinuity of the most representation for rotation space like Euler angle or quaternion will increase the unstable training process, we utilize rotation6d [67] to get stable *multi-scale* tokens. And for the distance function $\text{dist}(\cdot)$, we use cosine similarity rather than Euclidean distance which is the cause of unstable training based on our observations. In practice, due to the orthogonality between dimensions, we use a separate VQVAE for each dimension of the action space, with $V = 512$ and $C = 8$, to gain stable training. All convolutions we used in CARP are 1D to capture the time-dimension features.

Autoregressive. We adopt the architecture of standard decoder-only transformers akin to GPT-2 [12, 54]. The state s_o is used to generate the coarse first scale tokens and then is utilized as adaptive normalization [37] for the subsequent predictions. During the training process, we figure out that Exponential Moving Average (EMA) [19] will improve the training stability and the performance in around 4-5%, like [13]. During the inference, kv-caching can be used and no mask is needed.

4. Experiment

In this section, we evaluate the Coarse-to-fine Autoregressive Policy (CARP) across a range of widely used robotics tasks, including both state-based and image-based benchmarks. We assess CARP's performance in terms of task success rates and inference speed. Furthermore, we demonstrate CARP's practical effectiveness by deploying it on a real-world task with a UR5e robotic arm, comparing its performance to state-of-the-art diffusion-based models.

We organize our experiments to address the following key research questions:

- **RQ1**: Can CARP match the accuracy and robustness of current state-of-the-art diffusion-based policies?
- **RQ2**: Does CARP maintain high inference efficiency and achieve fast convergence?

4.1. Evaluation on Simulation Benchmark

We first evaluate CARP on a set of simulated tasks commonly used to benchmark diffusion-based models, assessing its ability to match their performance while significantly improving computational efficiency.

Figure 4. **Simulation Tasks.** We evaluate three tasks from the Robomimic benchmark [34]: Lift, Can, and Square, ordered by increasing difficulty from left to right.

Experimental Setup. We use the Robomimic [34] benchmark suite, which is widely adopted for evaluat-

Policy	Lift	Can	Square	Prams/M	Speed/s
BET [48]	0.96	0.88	0.54	**0.27**	**2.12**
DP-C [13]	**1.00**	0.94	**0.94**	65.88	35.21
DP-T [13]	**1.00**	**1.00**	0.88	8.97	37.83
CARP(Ours)	**1.00**	**1.00**	**0.98**	**0.65**	**3.07**

Table 1. **State-based Simulation Results (State Policy).** We report the average success rate of the top 3 checkpoints, along with model parameter scales and inference time for generating 400 actions. CARP significantly outperforms BET and achieves competitive performance with state-of-the-art diffusion models, while also surpassing DP in terms of model size and inference speed.

Policy	Lift	Can	Square	Prams/M	Speed/s
IBC [17]	0.72	0.02	0.00	**3.44**	32.35
DP-C [13]	**1.00**	0.97	**0.92**	255.61	47.37
DP-T [13]	**1.00**	**0.98**	0.86	9.01	45.12
CARP(Ours)	**1.00**	**0.98**	**0.88**	**7.58**	**4.83**

Table 2. **Image-based Simulation Results (Visual Policy).** Results show that CARP consistently balances high performance and high efficiency. We highlight our results in light-blue.

ing diffusion-based models [13, 39, 57]. We evaluate CARP on both state-based and image-based datasets collected from expert demonstrations, with each task containing 200 demonstrations. The selected tasks—Lift, Can, and Square—are standard in single-task evaluations, as shown in Fig. 4.

Baselines. We compare CARP with previous autoregressive policies as well as recent diffusion policies. Behavior Transformer (BET) [48] is an autoregressive model with action discretization and correction mechanisms, similar to offset-based prediction. Implicit Behavior Cloning (IBC) [17] utilizes energy-based models for supervised robotic behavior learning. Diffusion Policy (DP) [13] combines a denoising process with action prediction, implemented in two variants: CNN-based (DP-C) and Transformer-based (DP-T).

Metrics. For each task, we evaluate the policies by running 50 trials with random initializations to compute the success rate and report the average success rate across the top 3 checkpoints. Inference speed is measured on an A100 GPU as the average time for 400 action predictions over 5 runs to ensure robustness. We also record the parameter count for each model using the same PyTorch implementation interface.

Implementation Details. For baseline models, we follow the same implementation and training configurations provided by [13]. In state-based experiments, we set the observation horizon $o = 2$ and the prediction horizon $H = 16$ across all models. For image-based experiments, we set

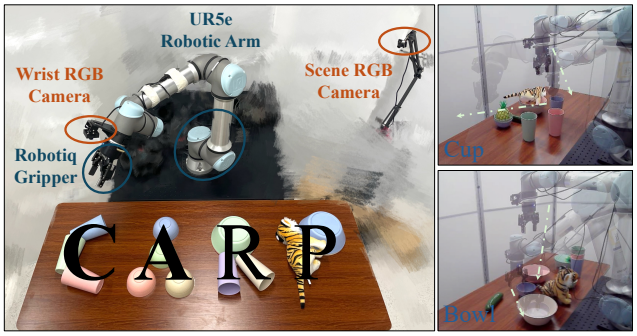
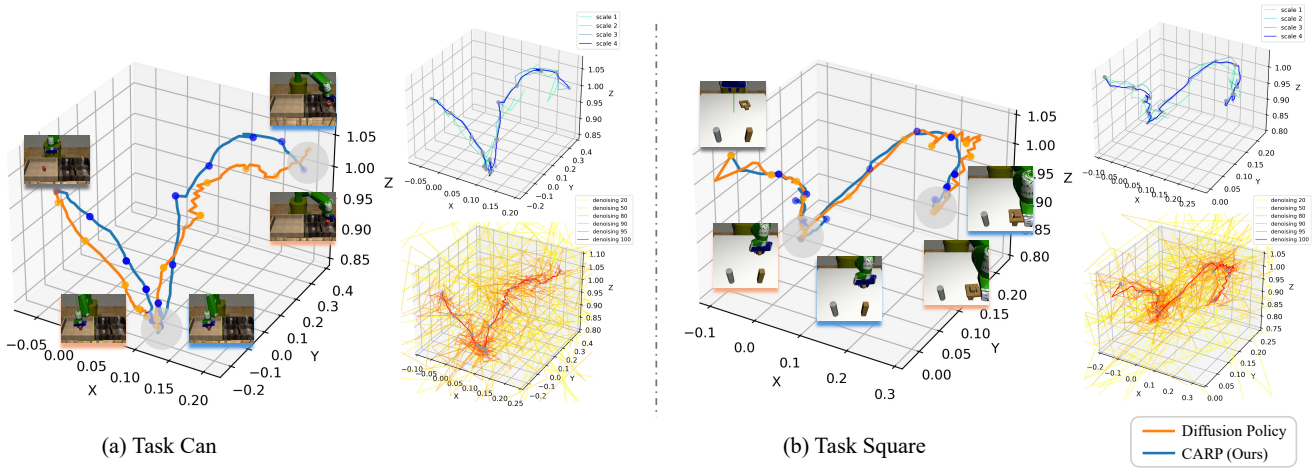


Figure 5. **Real-world Setup.** The left panel shows the environment used for the experiment and demonstration collection. The right panel shows the trajectory from the Cup and Bowl datasets.

$o = 1$ and $H = 16$ for better transferability to real-world scenarios. As per the benchmark, only the first 8 actions in the prediction horizon are executed starting from the current step. It is worth noting that our formulation includes a minor discrepancy in the horizon definition: the horizon H also covers the past observed steps, meaning the index of the current next predicted action is o instead of 0. This implementation detail, introduced by Diffusion Policy [13], is retained for consistency in comparisons. For CARP, we first train an action VQVAE model (see Sec. 3.1) similar to [27] , with a batch size of 256 and 300 epochs per task. Given a horizon $H = 16$, we design multi-scale representations with scales of $1, 2, 3$ and 4 to capture coarse-to-fine information across the action sequence. We then train an autoregressive GPT-2 style, decoder-only transformer (see Sec. 3.2), based on [54], using the same training settings as the benchmark, with a batch size of 256 for state-based experiments (4000 epochs) and a batch size of 64 for image-based experiments (3000 epochs).

Results. As shown in Tables 1 and 2, CARP consistently achieves comparable performance to state-of-the-art diffusion models across both state-based and image-based tasks, answering **RQ1**. Notably, CARP significantly outperforms diffusion policies in terms of inference speed, being approximately 10 times faster, with only 1-5% of the parameters required by diffusion models, thereby strongly supporting **RQ2** regarding CARP's efficiency. Leveraging its GPT-style autoregressive design [9, 44], CARP can seamlessly adapt to multi-task settings without complex design modifications [57], demonstrating the flexibility of this architecture (See Supp. for details).

Analysis. To further examine CARP's stability and efficiency, we visualize spatial trajectories along the xyz axes for the Can and Square tasks in Fig. 6. In each task's left panel, CARP consistently reaches specific regions (light grey area) to perform task-related actions, such as positioning for object grasping or placement. Com-



(a) Task Can (b) Task Square

Figure 6. **Visualization of the Trajectory and Refining Process.** The left panel shows the final predicted trajectories for each task, with CARP producing smoother and more consistent paths than Diffusion Policy (DP). The right panel visualizes intermediate trajectories during the refinement process for CARP (top-right) and DP (bottom-right). DP displays considerable redundancy, resulting in slower processing and unstable training, as illustrated by 6 selected steps among 100 denoising steps. In contrast, CARP achieves efficient trajectory refinement across all 4 scales, with each step contributing meaningful updates.

pared to diffusion-based models, CARP's trajectories are smoother and more consistent, underscoring its stability and accuracy (supporting **RQ1**). The right panels of Fig. 6 compare CARP's *coarse-to-fine* action predictions with 6 selected denoising steps of the diffusion model. CARP achieves accurate predictions within just 4 *coarse-to-fine* steps, whereas the diffusion model requires numerous denoising iterations, with many early steps introducing redundant computations. This analysis further supports **RQ2** and highlights CARP's efficiency and fast convergence advantage over diffusion policies in generating accurate actions with fewer refinement steps, as detailed in Sec. 3.2.

4.2. Evaluation on Real-world

In this section, we evaluate our approach, CARP, on real-world tasks under compute-constrained conditions, comparing its performance and efficiency against baseline methods.

Experimental Setup. To validate CARP's real-world applicability, we design two manipulation tasks:

1. *Cup*: The robot must locate a cup on the table, pick it up, and tilt it to a position suitable for human handover (top-right, Fig. 5).
2. *Bowl*: The robot needs to identify a smaller bowl and a larger pot on the table, pick up the bowl, and place it inside the pot (bottom-right, Fig. 5).

We use a UR5e robotic arm with a Robotiq-2f-85 gripper, equipped with two RGB cameras: one mounted on the wrist and one in a third-person perspective (left panel, Fig. 5). The robot is controlled through 6D end-effector positioning, with inverse kinematics for joint angle calculation. We collected 60 human demonstration trajectories for each task using a 3D Connexion space mouse for teleoperation.

Baselines. As a baseline, we reproduce the CNN-based visual diffusion policy [13], adapting the model's input size to accommodate our observational setup.

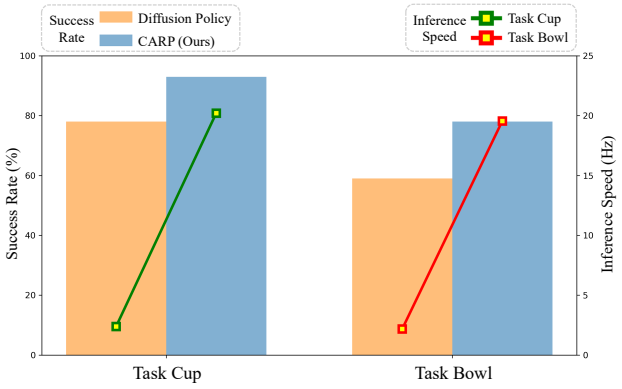


Figure 7. **Real-world Results (Visual Policy).** We report the average success rate across 20 trials and the inference speed as action prediction frequency. CARP achieves competitive success rates with significantly faster inference compared to diffusion policies.

Metrics. For each trained policy, we report the average success rate across 20 trials per task, with the initial positions randomized. We also measure inference speed on an NVIDIA GeForce RTX 2060 GPU, reporting action prediction frequency in Hertz.

Implement Details. For both models, the input consists of current visual observations from the wrist and scene cameras (resolution: 120×160), as well as proprioceptive data from the robotic arm. We execute 8 predicted actions out

of a horizon of 16 predictions. We train the diffusion policy for 3000 epochs with a batch size of 64. For CARP, we use the same visual policy structure as in the simulation tasks, training the action VQVAE for 300 epochs with a batch size of 256, and the decoder-only transformer for 3000 epochs with a batch size of 64.

Results. As shown in Fig. 7 and Fig. 8, CARP achieves comparable or superior performance, with up to a 10% improvement in success rate over the diffusion policy across all real-world tasks, supporting **RQ1**. Additionally, CARP achieves approximately 8× faster inference than the baseline on limited computational resources, demonstrating its suitability for real-time robotic applications, thus supporting **RQ2**.

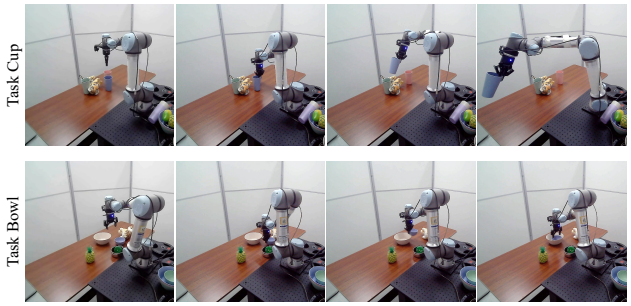


Figure 8. **Visualization of CAPR on Real-world Tasks.** CARP generates smooth and successful trajectories for the Cup and Bowl tasks, with temporal progression from left to right.

5. Related Work

5.1. Visual Generation

Advancements in generative models for visual generation have significantly influenced the robotics community. Autoregressive models generate images in a raster-scan fashion using discrete tokens from image tokenizers [15, 21] VQGAN [15] adopts vector quantization (VQVAE [56]) to discretize image features. GPT-2-style transformers [4, 27, 45, 59] demonstrate strong performance by generating tokens sequentially. Recent work [11, 60] has scaled these models to billions of parameters, achieving impressive text-to-image synthesis results. Diffusion models [50] generate images by progressively adding noise and training models like U-Net [47] to reverse the process. Recent improvements [36, 51] have enhanced their efficiency. Foundational models like Stable Diffusion 3.0 [16], SORA [8], and Vidu [5] continue to push the boundaries of high-quality image generation. VAR [54] introduces a new next-scale autoregressive paradigm that shifts image representation from patches to scales. This framework [54], has been applied across tasks [18, 29, 30, 33, 40, 63, 64]. Studies [53, 54] demonstrate that autoregressive models can achieve state-of-the-art performance compared with diffusion models.

5.2. Visuomotor Policy Learning

Autonomous decision-making in robots, without explicit programming, remains a core challenge [10]. Behavior cloning has shown promise, particularly in autonomous driving [38] and manipulation [65], in contrast to complex reinforcement learning approaches. Explicit policies directly map states or observations to actions [65], enabling efficient inference. However, they struggle with complex tasks. Techniques like action space discretization [62] and Mixture Density Networks (MDNs) [14, 48] have been proposed, though they face issues with exponential action space growth and hyperparameter sensitivity. Implicit policies, often based on Energy-Based Models (EBMs) [17, 23], offer flexibility but are challenging to train due to optimization instabilities. Diffusion models have been applied to robotic policy learning [2, 13, 46], showing their effectiveness for decision-making tasks. However, they are computationally expensive due to the multi-step denoising process. Subsequent work focuses on improving generalization [58], adapting to 3D environments [57, 61], and enhancing modularity with Mixture of Experts (MoE) [57]. Consistency models [32, 52] have been proposed to accelerate inference, though at the cost of increased model complexity.

6. Conclusion

In this work, we introduce Coarse-to-fine Autoregressive Policy (CARP), a novel paradigm for robotic visuomotor policy learning that combines the efficiency of autoregressive modeling (AM) with the high performance of diffusion modeling (DM). CARP incorporates: 1) multi-scale action tokenization to capture global structure and temporal locality, addressing AM's limitations in long-term dependency; 2) coarse-to-fine autoregressive prediction that refines actions from high-level intentions to detailed execution, achieving DM-like performance with AM-level efficiency through latent space prediction and relaxed Markovian constraints. The comprehensive evaluations, from simulation to real-world scenarios, demonstrate CARP's effectiveness in balancing high performance and efficiency.

We hope this work will inspire further exploration into next-generation policy learning by leveraging GPT-style autoregressive models, advancing a more unified perspective on current generative modeling techniques.

Limitations and Future Work. Our model's complexity is influenced by multi-scale action encoding and a two-stage training process. Future work could explore more effective action representations and broader GPT-style capabilities, such as zero-shot generalization and scaling laws based on CARP's frame. Additionally, expanding CARP to incorporate tactile and auditory inputs through a unified, decoder-only model could enhance its versatility across a wide range of robotic tasks.

References

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. 2

[2] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022. 8, 1

[3] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. 2

[4] Dosovitskiy Alexey. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv: 2010.11929*, 2020. 8

[5] Fan Bao, Chendong Xiang, Gang Yue, Guande He, Hongzhou Zhu, Kaiwen Zheng, Min Zhao, Shilong Liu, Yaole Wang, and Jun Zhu. Vidu: a highly consistent, dynamic and skilled text-to-video generator with diffusion models. *arXiv preprint arXiv:2405.04233*, 2024. 8

[6] Mariusz Bojarski. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016. 5

[7] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. 2

[8] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. 2024. *URL https://openai. com/research/video-generation-models-as-world-simulators*, 3, 2024. 8

[9] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 6, 2

[10] Harish chaandar Ravichandar, Athanasios S. Polydoros, Sonia Chernova, and Aude Billard. Recent advances in robot learning from demonstration. *Annu. Rev. Control. Robotics Auton. Syst.*, 3:297–330, 2020. 8

[11] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023. 8

[12] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023. 5

[13] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023. 1, 2, 3, 5, 6, 7, 8

[14] ZJ Cui, Y Wang, NMM Shafiullah, and L Pinto. From play to policy: Conditional behavior 423 generation from uncurated robot data. *arXiv preprint arXiv:2210.10047*, 424, 2022. 1, 8

[15] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 8

[16] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. 8

[17] Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Proceedings of the 5th Conference on Robot Learning*, pages 158–168. PMLR, 2022. 6, 8

[18] Jiatao Gu, Yuyang Wang, Yizhe Zhang, Qihang Zhang, Dinghuai Zhang, Navdeep Jaitly, Josh Susskind, and Shuangfei Zhai. Dart: Denoising autoregressive transformer for scalable text-to-image generation. *arXiv preprint arXiv:2410.08159*, 2024. 3, 5, 8

[19] David Haynes, Steven Corns, and Ganesh Kumar Venayagamoorthy. An exponential moving average algorithm. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE, 2012. 5

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020. 2, 3

[21] Mengqi Huang, Zhendong Mao, Zhuowei Chen, and Yongdong Zhang. Towards accurate image coding: Improved autoregressive image generation with dynamic vector quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22596–22605, 2023. 8

[22] Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, 2022. 3, 4

[23] Daniel Jarrett, Ioana Bica, and Mihaela van der Schaar. Strictly batch imitation learning by energy-based distribution matching. *Advances in Neural Information Processing Systems*, 33:7354–7365, 2020. 8

[24] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 2

[25] Maciej Kilian, Varun Jampani, and Luke Zettlemoyer. Computational tradeoffs in image synthesis: Diffusion, masked-token, and next-token prediction. *arXiv preprint arXiv:2405.13218*, 2024. 1, 2

[26] 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. 2

[27] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11523–11532, 2022. 4, 6, 8

[28] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*, 2023. 2

[29] Xiang Li, Hao Chen, Kai Qiu, Jason Kuen, Jiuxiang Gu, Bhiksha Raj, and Zhe Lin. Imagefolder: Autoregressive image generation with folded tokens. *arXiv preprint arXiv:2410.01756*, 2024. 8

[30] Xiang Li, Kai Qiu, Hao Chen, Jason Kuen, Zhe Lin, Rita Singh, and Bhiksha Raj. Controlvar: Exploring controllable visual autoregressive modeling. *arXiv preprint arXiv:2406.09750*, 2024. 8

[31] Zhixuan Liang, Yao Mu, Hengbo Ma, Masayoshi Tomizuka, Mingyu Ding, and Ping Luo. Skilldiffuser: Interpretable hierarchical planning via skill abstractions in diffusion-based task execution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16467–16476, 2024. 1

[32] Guanxing Lu, Zifeng Gao, Tianxing Chen, Wenxun Dai, Ziwei Wang, and Yansong Tang. Manicm: Real-time 3d diffusion policy via consistency model for robotic manipulation. *arXiv preprint arXiv:2406.01586*, 2024. 5, 8

[33] Xiaoxiao Ma, Mohan Zhou, Tao Liang, Yalong Bai, Tiejun Zhao, Huaian Chen, and Yi Jin. Star: Scale-wise text-to-image generation via auto-regressive representations. *arXiv preprint arXiv:2406.10797*, 2024. 8

[34] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *arXiv preprint arXiv:2108.03298*, 2021. 5, 1

[35] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *7th Annual Conference on Robot Learning*, 2023. 1

[36] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 8

[37] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2337–2346, 2019. 5

[38] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988. 8

[39] 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. 6

[40] Kai Qiu, Xiang Li, Hao Chen, Jie Sun, Jinglu Wang, Zhe Lin, Marios Savvides, and Bhiksha Raj. Efficient autoregressive audio modeling via next-scale prediction. *arXiv preprint arXiv:2408.09027*, 2024. 8

[41] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2

[43] Ilija Radosavovic, Baifeng Shi, Letian Fu, Ken Goldberg, Trevor Darrell, and Jitendra Malik. Robot learning with sensorimotor pre-training. *arXiv:2306.10007*, 2023. 2

[44] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020. 6

[45] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019. 8

[46] Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*, 2023. 8

[47] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 8

[48] Nur Muhammad Mahi Shafiullah, Zichen Jeff Cui, Ariuntuya Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. In *Thirty-Sixth Conference on Neural Information Processing Systems*, 2022. 1, 2, 5, 6, 8

[49] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017. 1

[50] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 8

[51] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 8

[52] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023. 8

[53] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024. 8

[54] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905*, 2024. 3, 4, 5, 6, 8

[55] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018. 5

[56] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 3, 8

[57] Yixiao Wang, Yifei Zhang, Mingxiao Huo, Ran Tian, Xiang Zhang, Yichen Xie, Chenfeng Xu, Pengliang Ji, Wei Zhan, Mingyu Ding, et al. Sparse diffusion policy: A sparse, reusable, and flexible policy for robot learning. *arXiv preprint arXiv:2407.01531*, 2024. 1, 2, 6, 8

[58] Jingyun Yang, Zi-ang Cao, Congyue Deng, Rika Antonova, Shuran Song, and Jeannette Bohg. Equibot: Sim (3)-equivariant diffusion policy for generalizable and data efficient learning. *arXiv preprint arXiv:2407.01479*, 2024. 8

[59] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021. 8

[60] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022. 8

[61] 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 *ICRA 2024 Workshop on 3D Visual Representations for Robot Manipulation*, 2024. 5, 8

[62] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pages 726–747. PMLR, 2021. 8

[63] Jinzhi Zhang, Feng Xiong, and Mu Xu. G3pt: Unleash the power of autoregressive modeling in 3d generation via cross-scale querying transformer. *arXiv preprint arXiv:2409.06322*, 2024. 8

[64] Qian Zhang, Xiangzi Dai, Ninghua Yang, Xiang An, Ziyong Feng, and Xingyu Ren. Var-clip: Text-to-image generator with visual auto-regressive modeling. *arXiv preprint arXiv:2408.01181*, 2024. 8

[65] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 5628–5635. IEEE, 2018. 8

[66] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023. 1, 2, 5

[67] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019. 5

[68] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, Yifeng Zhu, and Kevin Lin. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020. 1

CVPR #3951

CARP: Visuomotor Policy Learning via Coarse-to-Fine Autoregressive Prediction

Supplementary Material

7. Evaluation on Multi-task Benchmark

We evaluate CARP on the MimicGen [35] multi-task simulation benchmark, widely used by the state-of-the-art Sparse Diffusion Policy (SDP) [57], to demonstrate CARP's flexibility, a result of its GPT-style autoregressive design.

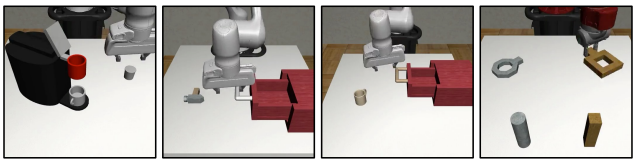


Figure 9. **Multi-task Simulation Setup.** We evaluate four tasks from the MimicGen benchmark: Coffee, Hammer Cleanup (Hammer), Mug Cleanup (Mug), and Nut Assembly (Nut), which are listed from left to right.

Experimental Setup. MimicGen extends benchmark Robomimic [34] by including 1K–10K human demonstrations per task, with diverse initial state distributions for enhanced generalization in multi-task evaluation. It consists of 12 robosuite [68] tasks powered by MuJoCo and 4 high-precision tasks from Isaac Gym Factory. We select 4 robosuite tasks—Coffee, Hammer, Mug, and Nut—for evaluation, each with 1K training trajectories, following the settings in SDP [57]. Visualization is shown in Fig. 9.

Baselines. We compare CARP with two baselines: Task-Conditioned Diffusion (TCD) [2, 31]: A basic diffusion-based multi-task policy and Sparse Diffusion Policy(SDP) [57]: A transformer-based diffusion policy that leverages Mixture of Experts (MoE) [49]. Both baselines are trained using visual inputs.

Metrics. Success rates are reported for each task as the average of the best three checkpoints. For Nut Assembly, partial success (e.g., placing one block inside the cylinder) is assigned a score of 0.5, while full success is scored as 1. For all other tasks, a score of 1 is awarded only when strict success criteria are fully satisfied. Additionally, we calculate the average success rate across all tasks. To evaluate model efficiency, we report both the parameter scale calculated by the Pytorch interface and the inference time required to predict 400 actions on a single A100 GPU.

Implementation Details. For CARP, we extend the single-task implementation by incorporating a task embedding as an additional condition, alongside the state or observation sequence s. We also use a moderately deeper decoder-only transformer in GPT-2 style. CARP is trained with a batch size of 512 for 100 epochs on an A100 GPU. Baseline models follow the same training settings as SDP [57]. This minimal modification enables CARP to adapt to multi-task learning seamlessly.

Policy	Coffee	Hammer	Mug	Nut	Avg.
TCD [31]	0.77	0.94	0.56	0.49	0.68
SDP [57]	0.82	**1.00**	**0.62**	0.54	0.75
CARP(Ours)	**0.92**	**1.00**	**0.62**	**0.66**	**0.80**

Table 3. **Multi-Task Simulation Results (Visual Policy).** Average success rates for each task are calculated from the best three checkpoints, along with the overall average success rate across all tasks. CARP demonstrates superior performance compared to existing diffusion-based policies.

Policy	Prams/M	Speed/s
SDP [57]	284.51	112.39
CARP(Ours)	**37.47**	**6.92**

Table 4. **Model Efficiency.** CARP demonstrates significantly lower parameter count and over 10× faster inference speed than SDP. Our results are highlighted in light-blue.

Results. As shown in Tab. 3, CARP achieves up to a 22% improvement in success rates compared to state-of-the-art diffusion-based models, highlighting its strong performance. With minimal modification, CARP seamlessly transitions from single-task to multi-task learning, further demonstrating its flexibility, a benefit of its GPT-style architecture. Additionally, as shown in Tab. 4, CARP achieves over 10× faster inference speed and uses only 10% of the parameters compared to SDP. These results strongly demonstrate CARP's efficiency, solidifying its role as a high-performance and high-efficiency approach for visuomotor robotic policies.

8. Limitations and Future Work

In this work, we propose CARP, a next-generation paradigm for robotic visuomotor policy learning, which effectively balances the long-standing trade-off between high performance and high efficiency seen in previous autoregressive modeling (AM) and diffusion modeling (DM) approaches. Despite these advancements, there remain several limitations and opportunities for improvement in future research.

First, the model structure of CARP can be further optimized for simplicity. Currently, CARP employs a two-stage

design, where the first stage utilizes separate *multi-scale* action VQVAE modules for each action dimension to address their orthogonality. Future work could focus on developing a unified *multi-scale* representation paradigm that directly integrates into the *coarse-to-fine* prediction process, enabling a more streamlined and efficient framework without compromising performance.

Second, the multimodal capabilities of CARP remain underdeveloped. To mitigate the unimodality issue inherent in traditional autoregressive policies trained with MSE loss, CARP adopts a cross-entropy loss, which retains the potential for multimodality. Compared to the Diffusion Policy's multimodal ability [13] likely coming from DDPM's integration over a Stochastic Differential Equation [20], CARP's approach is relatively straightforward. Further enhancements are needed to fully realize and leverage its multimodal potential.

Third, CARP's adoption of the GPT-style paradigm opens up promising yet unexplored possibilities. Beyond the flexibility already demonstrated, the contextual understanding capabilities inherent in GPT-style architectures [41] suggest that CARP could be extended to support multimodal inputs [3, 42] like tactile and auditory information and address robotic tasks requiring long-term dependency reasoning [1]. Additionally, its potential for generalization in few-shot and zero-shot learning scenarios [9] represents a particularly valuable direction for future exploration.

Finally, but not exhaustively, the scaling properties of CARP represent a promising frontier. The scaling laws established in existing GPT-style models [24] could be seamlessly applied to CARP, suggesting that increasing network capacity and leveraging larger pre-training datasets could lead to substantial performance gains. Furthermore, recent advances in Vision-Language-Action (VLA) [7, 26, 28] models present an opportunity to integrate CARP into such frameworks, which could further validate its scalability and capabilities.