FastDP: Deployable Diffusion Policy for Fast Inference Speed

Chang Shi, Amy Zhang
{chang.shi,amy.zhang}@austin.utexas.edu

The University of Texas at Austin

Abstract

Diffusion Policies have become a popular framework for robot visuomotor learning due to their superiority in capturing multi-modal distributions. However, the typical UNet and Transformer backbones for the policy network are computationally expensive, making them prohibitive for deployment on real robot systems that require real-time decision-making. In this paper, we address the challenge of efficient policy generation through a new architecture design. We explore the usage of structured state space models (SSMs), specifically the Mamba architecture, in helping diffusion policy inference speed. We further accelerate the diffusion process by starting the sampling from a Gaussian distribution around the previous action chunk. We validate our model on Adroit, MetaWorld, and Dexart environments, and show that it has 95% fewer parameters than the diffusion model with Unet backbone and consumes 85% less computation, leading to 3.3x inference speedup yet achieves similar performance on long-horizon tasks.

1 Introduction

Given large amounts of teleoperated data and human demonstration data, learning from demonstration is a very common method for robot visuomotor learning to avoid learning from scratch. However, the execution plan for a robotics task can often have more than one solution, making the demonstration dataset inherently multimodal. Diffusion models are renowned for capturing multimodal distributions underlying image data and generating diverse images sampled from this distribution in vision applications. (Chi et al., 2023) successfully used diffusion models for implicit behavior cloning and achieved impressive performance on visuomotor control both in simulation and real-world robot manipulation tasks. However, diffusion policies have high computation cost due to the iterative sampling mechanism in diffusion models, causing them to suffer from inference latency. Since inference speed is critical for robotic tasks in order to make real-time control decisions, (Chi et al., 2023) remedied this latency through receding horizon control to keep the decision-making module on par with the robot execution frequency. A more efficient model with low policy generation latency is an urgent necessity.

There have been various efforts to make diffusion models more efficient and scalable. (Peebles & Xie, 2023) explore a new class of diffusion models using the transformer architecture Vaswani et al. (2017) to replace the commonly used U-Net backbone. (Dasari et al., 2024) further validated that without sufficient hyperparameter tuning, diffusion transformers usually suffer from unstable training. Recently, structured state space sequence models (SSMs) have been proposed for modeling long-range dependencies (Gu et al., 2021; Fu et al., 2022). These models can be interpreted as a combination of RNNs and CNNs. They keep a latent state representation as a memory bank and recurrently update it according to a state equation while using a kernel to convolve over the input

sequence. (Gu & Dao, 2023) then designed an end-to-end neural network architecture, Mamba, which was further modified for discrete modalities, allowing the model to selectively propagate or forget information depending on the current input token. SSMs have linear time complexity compared to the quadratic attention mechanism in transformer models, leading to faster inference. Further, the authors designed a hardware-aware parallelized algorithm, making it suitable for time-sensitive tasks like rapidly adaptive robot control, which has stringent requirements for real-time long-horizon control and planning.

Thus, we propose FastDP, a diffusion policy model that adapts the end-to-end SSM architecture with efficient attention modules for robotic behavior cloning, aiming at better performance and faster inference speed. We also introduce historical actions as the sampling prior for the diffusion process, further boosting the sampling speed. To summarize, our main contributions are:

1. We propose FastDP, a diffusion policy model with a state space backbone, which reduce the computational complexity by 85% compared to UNet backbone.
2. We further accelerate the inference speed 1.7 times through starting the diffusion sampling from a Gaussian distribution around the last chunk of actions instead of pure noise.
3. We prove that the architecture has robust performance on complex robot manipulation tasks.

2 Preliminaries

State Space Models. State space models (SSMs) are a classic in control theory for modeling dynamic systems via state variables. Modern advances in deep learning have revived them for sequence modeling, beginning with the Structured State Space Sequence (S4) model introduced by (Gu et al., 2021), which leverages a diagonal plus low-rank parametrization to achieve an efficient convolutional representation of long sequences. SSMs model continuous sequences as follows:

$$h'(t) = \boldsymbol{A}h(t) + \boldsymbol{B}x(t), \quad y(t) = \boldsymbol{C}h(t).$$

By applying zero-order hold method - holding the value of each discrete sample constant until the next sample is taken, and with a step size Δ, the state matrix $\mathbf{A}$ and $\mathbf{B}$ are converted into an approximation matrix $\overline{\mathbf{A}} = \exp(\mathbf{\Delta A})$ and $\overline{\mathbf{B}} = (\mathbf{\Delta A})^{-1}(\exp(\mathbf{\Delta A}) - \mathbf{I}) \cdot \mathbf{\Delta B}$, therefore the discrete sequences can be modeled as

$$h(t) = \overline{\mathbf{A}}h(t-1) + \overline{\mathbf{B}}x(t), \quad y(t) = \mathbf{C}h(t).$$

In this recurrent representation, the state matrix $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$ are dynamic. A convolution kernel $\overline{\mathbf{K}}$ is used to convert it into a convolutional representation, which can be computed very efficiently. Follow up work Mamba (Gu & Dao, 2023) further improves upon S4 with selective scan algorithm parallelizable on hardware.

Compared to the $\mathcal{O}(N^2 d)$ complexity of transformer models, SSMs can reduce the complexity to $\mathcal{O}(1)$ at inference time, thus gaining popularity in fields where inference speed is crucial. However, while previous work often compares SSMs with transformers, SSMs have been somewhat disjoint from attention mechanisms until (Dao & Gu, 2024) showed the connection between the two – for the SSM $y = Mx$, there is a lower-triangular mask L such that $M = L \circ \left(\mathbf{C}\mathbf{B}^\top\right)$, which, if you rename $(\mathbf{C}, \mathbf{B}, x) \to (Q, K, V)$, is exactly causal linear attention:

$$Y = \left(L \circ QK^\top\right) V.$$

Essentially, state space models like Mamba Gu & Dao (2023) and other variants can be treated as a kind of linear attention-based method, and are naturally faster than softmax attention-based models like transformer.

Diffusion Models. Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) are a type of generative model that can capture the data distribution $q(\mathbf{x}_0)$ from a dataset by learning an inverse

process of the data gradually corrupted to pure noise. Diffusion models define (1) a forward noising process that gradually injects Gaussian noise over T discrete steps:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\big(\mathbf{x}_t;\ \sqrt{1-\beta_t}\,\mathbf{x}_{t-1},\ \beta_t\mathbf{I}\big),$$

where $\{\beta_t\}_{t=1}^{T} \in (0,1)$ is a noise schedule. The forward process is a fixed Markov chain, by multiplying along the chain, the marginal over $\mathbf{x}_t$ admits a closed-form:

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\big(\mathbf{x}_t;\ \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0,\ (1-\bar{\alpha}_t)\mathbf{I}\big),$$

where $\alpha_t = 1-\beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t}\alpha_s$. (2) a reverse process learnt to denoise $\mathbf{x}_t$ and recover the original data $\mathbf{x}_0$:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}\big(\mathbf{x}_{t-1};\ \mu_\theta(\mathbf{x}_t,t),\ \Sigma_\theta(\mathbf{x}_t,t)\big),$$

where $\mu_\theta(\mathbf{x}_t,t)$ and $\Sigma_\theta(\mathbf{x}_t,t)$ are the mean and covariance matrix at each step t. The model training is cast as minimizing a variational bound which, through careful choice of parameterization, reduces to a simple regression objective against known noise terms $\boldsymbol{\epsilon}$:

$$\mathcal{L} = \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\big[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t,t)\|^2\big],$$

where $\boldsymbol{\epsilon}_\theta$ is the noise prediction network. In the conditional case, an extra conditioning variable $\mathbf{c}$ can be added where the reverse process and the noise prediction network become $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{c})$ and $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c})$.

3 Related Work

Diffusion Based Policy Learning. Given the advanced capability of capturing multi-modality, diffusion-based methods have been popular for decision making. Decision Diffuser (Ajay et al., 2022) diffuses over the states and trains an inverse dynamics model for action extraction, while Diffuser (Janner et al., 2022) diffuses over both the states and actions. Diffusion Policy (DP)(Chi et al., 2023) instead diffuses on actions conditioned on states and focuses on robot deployment details. Building on top of these, Diff-Control (Liu et al., 2024) demonstrates that diffusion-based policies can acquire statefulness through a Bayesian formulation facilitated by ControlNet. More recently, to enhance visual generalization, the 3D Diffusion Policy (DP3) (Ze et al., 2024a) incorporates 3D point-cloud representations, furthur improving performance.

A brief comparison of diffusion-based policy learning frameworks is listed in Table 1. Our work is buit on top of the 3D diffusion policy but replace commonly used Unet and transformer architecture in the diffusion models with SSMs.

Table 1: A comparison of diffusion-based policy learning frameworks

Method	Diffusion Model	Characteristics
Diffuser (Janner et al., 2022)	$p(a_t, \ldots, s_{t+T-1}, a_{t+T-1} \mid s_t)$	Receptive field applied to ensure local consistency.
Decision diffuser (Ajay et al., 2022)	$p(s_{t+1}, \ldots, s_{t+T-1} \mid s_t, g_t)$	Inverse dynamics for future action selection.
Diffusion policy (Chi et al., 2023)	$p(a_t, \ldots, a_{t+T-1} \mid s_t, \ldots, s_{t-T+1})$	Deployed on real robots.
3D diffusion policy (Ze et al., 2024a)	$p(a_t, \ldots, a_{t+T-1} \mid s_t, \ldots, s_{t-T+1})$	Introduce point cloud modality.
Diff-Control (Liu et al., 2024)	$p(a_t, \ldots, s_t)$	Use ControlNet(Zhang et al.) as policy diffusion backbone.

Decision Making using State Space Models. Diffusion-based methods often suffer from slow inference. In order to meetrobot execution speed requirements, diffusion policy works like (Chi et al., 2023) tend to exclude observation features from the denoising process and use extra tricks like receding horizon control. Since the number of iterative sampling steps has the most impact on inference speed, developing more computationally efficient network architectures is a fruitful research direction.

Recently, there are an emerging number of works that start to use state-space models for decision making, either through imitation learning or reinforcement learning. (Jia et al., 2024) presented Mamba Imitation Learning (MAIL), using Mamba to mitigate the representation overfitting problem that transformers often suffer from. The authors explored encoder-only and encoder-decoder architecture and further verified the capability of MAIL in leveraging multi-modal inputs. Mamba policy(Cao et al., 2024a) also replaces the convolution block in Unet with Mamba in the imitation learning framework and showed improvement in inference speed. In the offline RL setting, Mamba decision maker(Cao et al., 2024b), Decision Mamba(Lv et al., 2024) and Decision Mamba-Hybrid (DM-H)(Huang et al., 2024) use Mamba to gather features at different granularity or hierarchically generate subgoals.

A brief comparison of policy learning frameworks that uses state space models is listed in Table 2. Our work has similar architecture but focusing on the introduction of warm up tricks to further increase the inference speed.

Table 2: A comparison of state space model accelerated policy learning frameworks

Method	Setting	SSM input	Characteristics
MAIL (Jia et al., 2024)	IL	s and a	Compare between encoder-only and encoder-decoder structures.
Mamba policy (Cao et al., 2024a)	IL	a	Retain Unet structure, replace the convolution blocks with Mamba.
Mamba decision maker (Cao et al., 2024b)	Offline RL	s, a and r	Similar structure as decision transformer, but replace transformer with local and global ssm branches to extract multi-scale features.
Decision mamba (Lv et al., 2024)	Offline RL	s,a and r	Concurrent work as Mamba decision maker.
Decision mamba (DM-H) (Huang et al., 2024)	Offline RL	s,a and terminal d	Hybrid model where Mamba is used to generate subgoals, transformer is used to generate actions.
FastDP(ours)	IL	a	

4 Methods

4.1 Action Sequence Prediction

Considering the common misalignment problem of model update frequency and robot execution frequency, we implement action sequence prediction with a multi-step horizon instead of single next-time step prediction. This could improve the smoothness of the robot policy and would especially benefit tasks that require long-horizon planning. We use three horizon parameters, T_o as the observation horizon, T_p as the prediction horizon, and T_a as the action horizon. At time step t, the policy takes as input T_o historical observations, that is, time step $t - T_o + 1$ to t. The policy predicts

an action sequence of length T_p, that is, step t to $t + T_p - 1$, among which the first T_a actions are executed before the next replanning.

4.2 Architecture Design

We use the DP3 Encoder, a lightweight MLP network designed in (Ze et al., 2024b) to process the point cloud inputs. It consists a three-layer MLP, a max-pooling function serve as an order-equivariant operation, and a final projection layer to project the point cloud features into a compact vector. After that, the concatenation of the robot state and point cloud representation forms the state feature used for the diffusion policy conditioning. Actions are first passed through an encoder, then Feature-wise Linear Modulation (FiLM)(Perez et al., 2018) is used to pass-in the conditioning input, where linear layer learns the affind transformation paramters a and b from the coniditoning input, which would element-wisely applied on to each entry of the action features. FiLM layer carries out a simple, feature-wise affine transformation on a neural network's intermediate features, and has been proved to be efficient in adaptive conditioning.The fused action features are passed into the Mamba state space model and 1D convolutional layer to predict action sequence T_p steps ahead. The entire network architecture is illustrated in Fig. 1.

4.3 Historical Action Warm Up

Given that robot actions are usually continuous, the current action is often similar to recent, previous actions. Therefore, it is not efficient to keep the robot execution timestep and diffusion timestep orthogonal by starting the sampling from pure noise to generate the next action sequence. We introduce a historical action chunk as the diffusion process prior, and start the reverse process by sampling from a narrow Gaussian distribution around the action chunk. Since the distribution distance between neighboring action sequences is likely smaller than the distance between an action sequence and pure Gaussian noise, the diffusion model can maintain high task performance while using fewer diffusion steps.

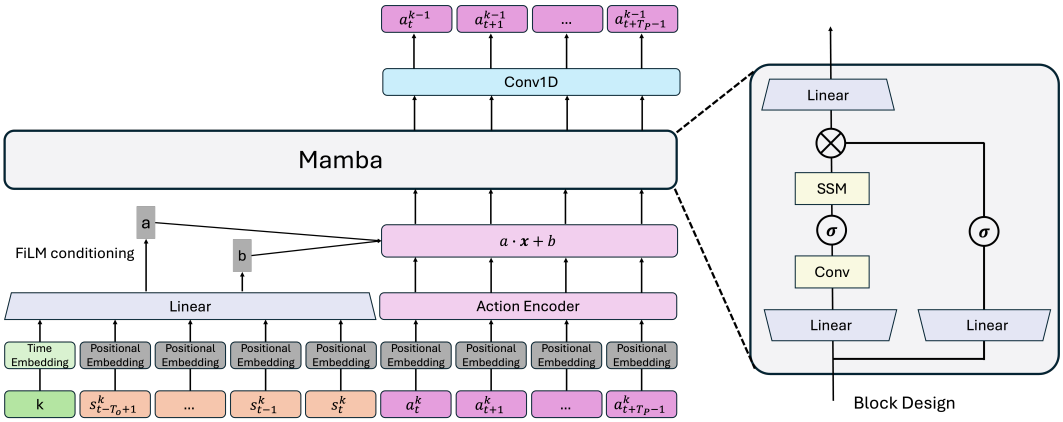

Figure 1: FastDP Network Architecture Overview

5 Experiments

We conduct experiments on multiple offline datasets, including Adroit (Rajeswaran et al., 2017), MetaWorld (Yu et al., 2020), and DexArt (Bao et al., 2023), some example tasks are illustrated in Fig. 2. We use the expert policy provided by (Ze et al., 2024a) to generate 10 episodes for each of the tasks and environments, which serves as the ground truth for imitation learning.

Baselines. The main focus of this work is to explore techniques that can increase the inference speed of the model and make diffusion-based policies more deployable to real robots. We use the

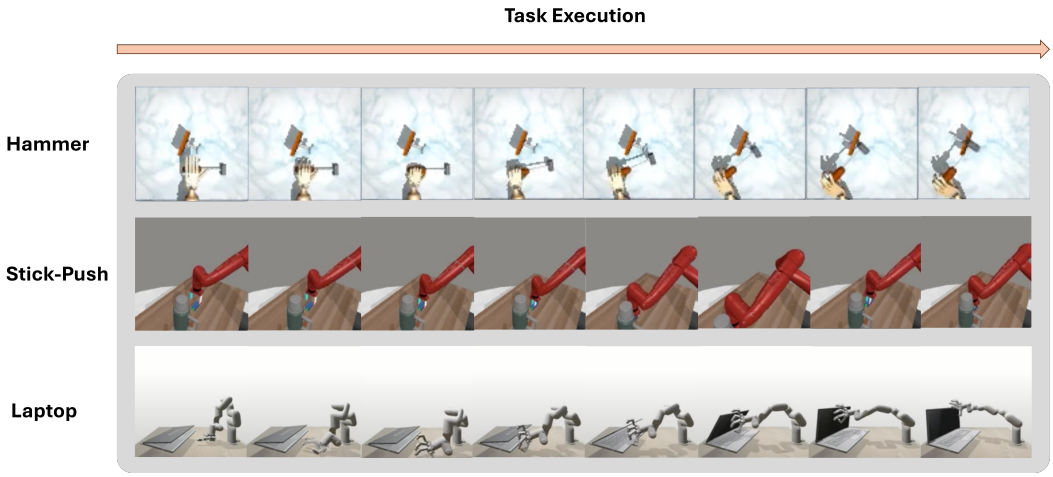

Figure 2: Visualization examples of manipulation Tasks, including Adorit Hammer, MetaWorld Stick-Push, and DexArt Laptop and many more.

diffusion policy (DP) and the 3D diffusion policy (DP3) as baselines. We aim for fewer model parameters and faster inference, while keeping the task success rates on par with these SOTA methods.

Implementation. In order to draw a fair comparison, we keep the training parameters consistent with those in DP3. We use Denoising Diffusion Implicit Models (DDIM) (Song et al., 2020), since it introduces a deterministic, non-Markovian process that allows for fewer sampling steps. The number of timesteps is set to 100 during training, and 10 during inference for the DDIM noise scheduler. The model is trained for a total of 3000 epochs with batch size of 128, linear layer hidden dimension of 128, and state space model dimension of 128. For historical action warm up, we use the action chunk of time $t - T_p$ to $t - 1$, and a standard deviation of 0.1 for the Gaussian distribution. We take $T_o = 2$, $T_p = 8$ and $T_a = 4$.

Evaluation metric. We run each experiment with three training seeds. For each seed, we evaluate 20 episodes every 200 training epochs and then compute the average of the highest 5 success rates. The mean and standard deviation of the success rates across 3 seeds are reported.

Further, we use the number of model parameters to measure the complexity. We use Giga Multiply-Add Operations per Second (GMACs) and Wall-clock time to compare the inference speed. Wall-clock time is reported using the average over 10 samples. All experiments are evaluated using a single NVIDIA RTX A5500 GPU.

5.1 Results

As illustrated in Table 3, SSMs can greatly reduce computation complexity, given that the computation of the backbone architecture is calculated each time step in the diffusion process, SSMs help reduce computational complexity by around 75%. While our model has significantly fewer parameters as shown in Table 4, the policy performance does not degrade but on par with other state-of-the-art diffusion models.

5.2 Number of diffusion steps

Since the warm-up variant of FastDP begins the diffusion process near the previous action sequence, the distribution difference between the initial samples and the goal action sequence is inherently smaller, leading to a natural thought to reduce the number of diffusion steps at inference time for further speed improvement. Therefore, we experiment FastDP w/ warmup using different number

Table 3: A comparison of the computational complexity of different backbones. Numbers are reported on Adriot Hammer task with prediction horizon $T_p = 8$. While Unet and FastDP w/o warmup are using 10 inference steps FastDP w/warmup is using 5 inference steps and have significant smaller computation latency.

Architecture	Unet	**FastDP w/o warmup (ours)**	**FastDP w/ warmup (ours)**
Model param #	$35.51M$	$\mathbf{1.80M}$	$1.95M$
Computation (GMACs)	49.18	12.34	$\mathbf{7.24}$
Wall-clock time(milliseconds)	52.66	27.32	$\mathbf{15.91}$

Table 4: A comparison of FastDP with other baselines in simulation. All models takes prediction horizon of $T_p = 8$, 100 steps at diffusion training time, and 10 steps at inference time, except for FastDP w/ warmup taking 5 steps at inference time.

Env	Task	DP	DP3	**FastDP w/o warmup**	**FastDP w/ warmup**
Adroit	Hammer	48 ± 17	$\mathbf{100 \pm 0}$	85 ± 2	87 ± 1
	Door	50 ± 5	$\mathbf{62 \pm 4}$	56 ± 11	56 ± 7
	Pen	24 ± 4	43 ± 6	47 ± 6	$\mathbf{54 \pm 2}$
MetaWorld	Assembly	15 ± 1	99 ± 1	79 ± 16	$\mathbf{100 \pm 0}$
	Disassemble	43 ± 7	69 ± 4	84 ± 7	$\mathbf{85 \pm 9}$
	Stick-Push	63 ± 3	97 ± 4	$\mathbf{100 \pm 0}$	95 ± 8
DexArt	Laptop	69 ± 4	$\mathbf{83 \pm 1}$	83 ± 4	69 ± 1
	Faucet	23 ± 8	$\mathbf{63 \pm 2}$	40 ± 2	34 ± 2
	Average	41.9	77.0	71.8	72.5

of diffusion steps at inference time, and the results are shown in Table. 5. As we can see, the model performance is very robust when we reduce the diffusion steps from 10 to 5 and even 3 during the inference time. Therefore, FastDP w/ warmup can achieve much faster inference speed than the w/o warmup version.

Table 5: A study of performance robustness with various diffusion steps during inference after using historical action warm up. The prediction horizon is set to $T_p = 8$. The result shows that there is minor degradation on task success rate when reducing the inference diffusion steps from 10 to 5 and 3.

Env	Task	steps = 3	steps=5	steps=10
Adroit	Hammer	83 ± 2	87 ± 1	88 ± 2
	Door	57 ± 5	56 ± 7	57 ± 7
	Pen	49 ± 4	54 ± 2	51 ± 3
MetaWorld	Assembly	100 ± 0	100 ± 0	99 ± 1
	Disassemble	87 ± 6	85 ± 9	82 ± 9
	Stick-Push	99 ± 1	95 ± 8	100 ± 0
DexArt	Laptop	67 ± 2	69 ± 1	71 ± 0
	Faucet	36 ± 3	34 ± 2	36 ± 2
	Average	72.3	72.5	73.0

5.3 Long-horizon Performance

Given that state-space models have principled mechanisms for modeling long-range dependencies, we expect that diffusion policy models with Mamba backbone would show more robust performance on long-horizon tasks than the ones with Unet backbone. Therefore, we tested different scale of prediction horizon $T_p = 8$, $T_p = 16$ and $T_p = 32$, the results are shown in Table. 6.

Table 6: A study of performance robustness of FastDP on long-horizon tasks. To validate the benefit of the state space model backbone, we run FastDP w/o warmup with 10 diffusion steps for inference at a predicition horizon of 8, 16 and 32. The results show that FastDP are good at long-horizon imitation learning.

Env	Task	$T_p = 8$	$T_p = 16$	$T_p = 32$
Adroit	Hammer	85 ± 2	81 ± 2	72 ± 4
	Door	56 ± 11	55 ± 4	56 ± 4
	Pen	47 ± 6	50 ± 3	49 ± 6
MetaWorld	Assembly	79 ± 16	96 ± 3	99 ± 1
	Disassemble	84 ± 7	84 ± 5	86 ± 2
	Stick-Push	100 ± 0	100 ± 0	100 ± 0
DexArt	Laptop	84 ± 4	59 ± 5	12 ± 7
	Faucet	40 ± 2	35 ± 1	33 ± 3
	Average	71.9	70.0	63.4

6 Limitation and future work

There are various new state space models and linear attention models emerging, including but not limited to the introduction of delta rule, gated mechanism, structured attention matrix designs. (Yang et al., 2024b;a; Liu et al.). In the future, we could analyze the benefit of each module and better understand how these models contribute to balancing between long-horizon memory and forgetting unimportant information, as well as improving inference speed. Meanwhile, state space models are closely related to classic control theories, naturally leading to a broad area where stability and sensitivity of the dynamics can be utilized as model constraints to better adapt the policy to the robot system, like Block et al. (2023). We leave these for future exploration.

7 Conclusion

State space models, as the backbone of diffusion policy, can greatly accelerate the inference speed, comparing to Unet and transformer-based policy models. Using historical action chunk as the diffusion sampling prior can further reduce the diffusion steps. FastDP shows that the reduction of model complexity did not leads to significant performance drop, making these new architecture greatly suitable for real robot deployment.

References

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.

Chen Bao, Helin Xu, Yuzhe Qin, and Xiaolong Wang. Dexart: Benchmarking generalizable dexterous manipulation with articulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21190–21200, 2023.

Adam Block, Ali Jadbabaie, Daniel Pfrommer, Max Simchowitz, and Russ Tedrake. Provable guarantees for generative behavior cloning: Bridging low-level stability and high-level behavior. *Advances in Neural Information Processing Systems*, 36:48534–48547, 2023.

Jiahang Cao, Qiang Zhang, Jingkai Sun, Jiaxu Wang, Hao Cheng, Yulin Li, Jun Ma, Yecheng Shao, Wen Zhao, Gang Han, et al. Mamba policy: Towards efficient 3d diffusion policy with hybrid selective state models. *arXiv preprint arXiv:2409.07163*, 2024a.

Jiahang Cao, Qiang Zhang, Ziqing Wang, Jingkai Sun, Jiaxu Wang, Hao Cheng, Yecheng Shao, Wen Zhao, Gang Han, Yijie Guo, et al. Mamba as decision maker: Exploring multi-scale sequence modeling in offline reinforcement learning. *arXiv preprint arXiv:2406.02013*, 2024b.

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

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

Sudeep Dasari, Oier Mees, Sebastian Zhao, Mohan Kumar Srirama, and Sergey Levine. The ingredients for robotic diffusion transformers. *arXiv preprint arXiv:2410.10088*, 2024.

Daniel Y Fu, Tri Dao, Khaled K Saab, Armin W Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models. *arXiv preprint arXiv:2212.14052*, 2022.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Sili Huang, Jifeng Hu, Zhejian Yang, Liwei Yang, Tao Luo, Hechang Chen, Lichao Sun, and Bo Yang. Decision mamba: Reinforcement learning via hybrid selective sequence modeling. *arXiv preprint arXiv:2406.00079*, 2024.

Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

Xiaogang Jia, Qian Wang, Atalay Donat, Bowen Xing, Ge Li, Hongyi Zhou, Onur Celik, Denis Blessing, Rudolf Lioutikov, and Gerhard Neumann. Mail: Improving imitation learning with selective state space models. In *8th Annual Conference on Robot Learning*, 2024.

H Liu, F Liu, X Fan, and D Huang. Polarized self-attention: Towards high-quality pixel-wise regression. arxiv 2021. *arXiv preprint arXiv:2107.00782*.

Xiao Liu, Yifan Zhou, Fabian Weigend, Shubham Sonawani, Shuhei Ikemoto, and Heni Ben Amor. Diff-control: A stateful diffusion-based policy for imitation learning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7453–7460. IEEE, 2024.

Qi Lv, Xiang Deng, Gongwei Chen, Michael Yu Wang, and Liqiang Nie. Decision mamba: A multi-grained state space model with self-evolution regularization for offline rl. *Advances in Neural Information Processing Systems*, 37:22827–22849, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. *arXiv preprint arXiv:2412.06464*, 2024a.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. *arXiv preprint arXiv:2406.06484*, 2024b.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

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 *Proceedings of Robotics: Science and Systems (RSS)*, 2024a.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv preprint arXiv:2403.03954*, 2024b.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models.