LEARNING LONG-CONTEXT ROBOT POLICIES VIA PAST-TOKEN PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Complex robotic tasks often require spatiotemporal reasoning over long sequences of actions and observations. Yet learning long-context policies remains difficult: as context length increases, the training process becomes increasingly compute and memory-intensive, and covariate shifts at deployment become more pronounced. Recent methods typically sidestep these challenges by discarding significant portions of the historical context, risking the loss of crucial information for subsequent decisions. In this paper, we propose a two-stage training approach that explicitly regularizes the information preserved in the learned representation: first, we pre-train a short-context encoder to predict a long sequence of future actions, maximizing the information each frame encodes about long-range dependencies; then, given pre-computed frame embeddings, we fine-tune a long-context decoder on an auxiliary task, where the policy learns to predict past actions alongside future ones. This simple design yields two surprising benefits: substantially reduces memory consumption during training and greatly improves history awareness of the learned policy. Additionally, the auxiliary task provides a natural mechanism for self-verification, allowing the policy to assess its sampled predictions at test time. Experiments on manipulation tasks that necessitate extensive historical context demonstrate that our proposed method improves the performance of long-context policy by 3× and accelerates policy training by more than 10×.

1 INTRODUCTION

Many robotic tasks are inherently non-Markovian: the optimal action at a given timestep may depend not only on the current observation but also on past observations and actions (Mandlekar et al., 2022; Zhao et al., 2023a; Lee et al., 2024b; Zheng et al., 2024). For example, consider manipulation tasks where the robot arm occludes critical parts of the scene, or multi-stage tasks where early steps inform later strategies (Nasiriany et al.). Likewise, past actions can prescribe a style of execution – such as speed, curvature, or strategy – that shapes how future actions should unfold (Chi et al., 2023b; Liu et al., 2024).

Despite the importance of history for decision-making, learning long-context policies through imitation learning remains difficult due to two fundamental challenges. First, incorporating more historical images into policy inputs often increases the presence of spurious correlations in training data, leading to amplified covariate shifts and degraded performance during deployment (Ross & Bagnell, 2010). Often, a policy that can solve a task well with low history degrades to near-zero performance when context length is increased. Second, conditioning on high-dimensional image sequences imposes a rapidly growing memory and computation burden, making end-to-end training practically infeasible at scale (Zheng et al., 2024; Li et al., 2024a).

To circumvent these challenges, recent methods often limit the amount of historical information the policy sees – either by truncating the context length (Chi et al., 2023b; Black et al., 2024) or by engineering past observations into compact representations, such as selecting key frames (Wen et al., 2021) and summing observations (Zheng et al., 2024)). While these strategies can make training more manageable, they may discard information critical for subsequent decisions. How can we enable a robot policy to learn more effectively and efficiently from a long sequence of visual observations?

In this paper, we introduce a simple and general recipe for learning long-context policies. At the core of approach is *past token prediction*: an auxiliary training objective, where the policy learns to reconstruct its previous actions alongside predicting future ones. We find that past token prediction

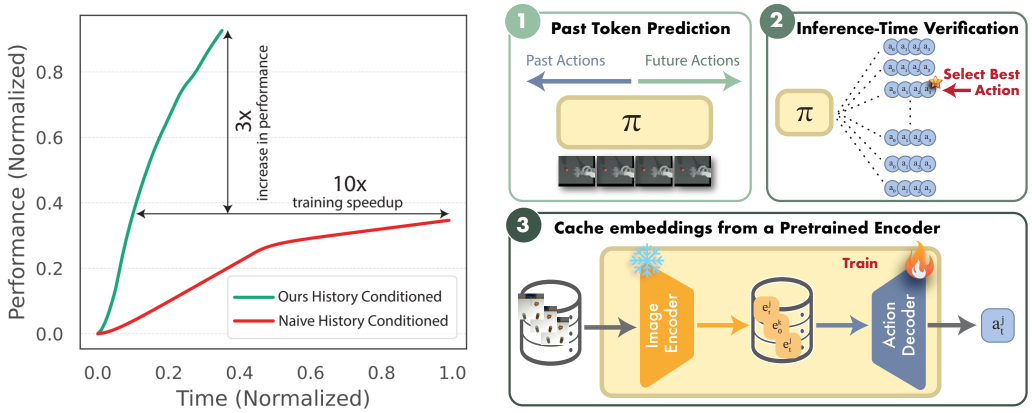

Figure 1: We propose a simple framework for learning long-context robot policies. Our method leads to 3x in performance while reducing the training expense by more than 10x.

enables the learned policy to more effectively attend to relevant information across its entire history, leading to substantial performance improvements. Crucially, we find that this performance gain stems primarily from improved decoder representations, rather than improved feature extraction in single-step observation encoding.

Motivated by this observation, we propose to train a long-context robot policy in two stages. First, we pre-train a short-context encoder to predict an extended sequence of future actions. Then, we fine-tune a long-context decoder that jointly predicts past and future actions from pre-computed frame embeddings. This design enables the policy to capture long-range temporal dependencies without incurring prohibitive memory costs. Furthermore, past token prediction naturally enables the learned policy to self-verify: by comparing its predicted past actions with those actually taken, the policy can estimate the quality of each sampled output and selectively execute the most reliable one at test time.

Our main contributions are two-fold: (i) revisit the effect of past-token prediction on long-context robot policy (§3.1), (ii) propose a two-stage training recipe to boost training efficiency (§3.2). Empirically, we validate our method on transformer-based polices across eight challenging simulation and real-world tasks (§4). Our results show that the proposed method increases the success rate of long-context policies by 3x on average while reducing training overhead by more than 10 times. Notably, our method enables policies to solve extended-horizon tasks at 80% success rate, where prior approaches fail entirely.

2 Related Work

Imitation Learning. Imitation learning has long served as a simple yet powerful paradigm for robot learning (Argall et al., 2009; Ravichandar et al., 2020; Zare et al., 2024). Early approaches typically framed it as a supervised learning problem, where the policy learns to map a given observation to the target action (Ross & Bagnell, 2010). More recent works have shifted toward modeling the distribution of demonstrations (Zhao et al., 2023b; Chi et al., 2023a; Lee et al., 2024a; Ze et al., 2024; Bharadhwaj et al., 2024; Wang et al., 2024; Haldar et al., 2024; Liu et al., 2024).

This approach has recently achieved remarkable success towards generalist robot policies (Black et al., 2024), with performance improving as the scale of training data grows (Li et al., 2024b). Nevertheless, imitation learning remains highly susceptible to covariate shift (de Haan et al., 2019; Wen et al., 2020b): works like (Ross et al., 2011) and (Spencer et al., 2021) characterize compounding errors in a feedback loop once the learned policy diverges from the demonstration manifold. This problem is exacerbated by high-dimensional visual inputs where less-robust features might be learned due to overspecification (Nasiriany et al., 2024). Notably, recent works (Chi et al., 2023b; Zheng et al., 2024) have empirically found that image-conditioned specialist and generalist policies to degrade with history. Our work introduces a training recipe that counteracts this effect while improving efficiency, aiding the development of future history-conditioned imitation learning policies.

Long-Context Policies. Many works have tried to cope with learning over extended sequences of high-dimensional observations by compressing observations in a more robust form. For example,

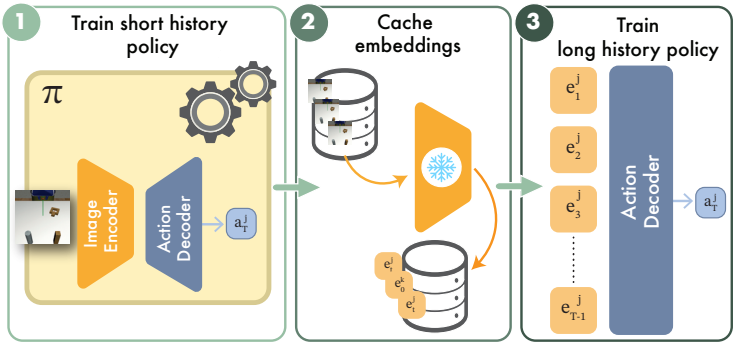

Figure 2: Overview of two-stage training with embedding caching. As PTP acts on the decoder, caching embeddings substantially improves inference speed without sacrificing performance. We use a visual encoder from a short-range policy with low validation loss to compute the embeddings of the images in the buffer and cache them in the buffer. With the cached embeddings we can train the long-horizon policy much faster. At test time we take the original encoder.

early work studying "copycat" behavior often employed specialized regularizers – such as adversarial objectives (Wen et al., 2020a) and information bottlenecks (Seo et al., 2023) – to remove past action information. Recent works also use features like keyframes (Wen et al., 2021) and motion tracks (Ren et al., 2025). Strategies like sketch synthesis (Sundaresan et al., 2024) and visual trace prompting (Zheng et al., 2024) have been implemented in generalist robot policies for similar purposes.

While these techniques can be effective under specific scenarios, they inherently make assumptions that will not hold across tasks such as discarding portions of the input sequence that might be critical for subsequent decisions. In contrast, our approach regularizes the policy through an auxiliary objective – past token prediction – that yields a reliable representation in the history decoder without explicitly discarding features. Most closely related to our work, BC-RNN (Mandlekar et al., 2022), Diffusion Policy (Chi et al., 2023b) and VQ-BeT (Lee et al., 2024b) implicitly incorporate this idea. We systematically revisit this technique, introduce a two-stage training process that substantially reduces training time and memory footprint, and propose a test-time verification mechanism, further improving performance in challenging settings.

Inference-Time Scaling. Recent research in language modeling, image generation, and robotics have shown that inference-time compute may allow models to improve their performance (Bansal et al., 2024; Ma et al., 2025; Nakamoto et al., 2024). Some seeks to build an additional verifier to re-rank the output samples (Cobbe et al., 2021; Weng et al., 2023b; Lightman et al., 2023; Yu et al., 2024), while others propose to leverage on the internal knowledge to improve reasoning through self-verification (Weng et al., 2023a; Stechly et al., 2024). Our method echoes the latter paradigm in the robotic context: our policy is trained to predict accurate past actions before predicting the present action and can self-verify at test-time through past action accuracy. Similar to how it may be more compute-efficient to use test-time compute on a small LLM (Snell et al., 2024), we show checkpoints trained for fewer epochs or at shorter histories can approach the performance of optimal checkpoints by using more test-time compute.

3 METHOD

Learning robot policies conditioned on long observation histories has been a long-standing challenge due to two key factors: high memory demands during training and compounding covariate shifts at deployment. In this section, we introduce a learning approach that explicitly regularizes internal representations to improve both training efficiency and policy robustness. Specifically, we will first revisit past-token prediction, an auxiliary task that has been implicitly used in recent policies but largely underexplained in the literature (§3.1). Building on our analysis, we will then present a two-stage training recipe that preserves the benefits of this auxiliary task while reducing memory consumption (§3.2). Finally, we will introduce an inference technique that leverages on the auxiliary task to effectively self-verify sample outputs at test time (§3.3).

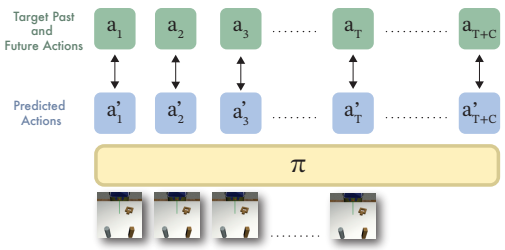

Figure 3: Illustration of past-token prediction. The transformer-based decoder jointly predicts both past and future action tokens.

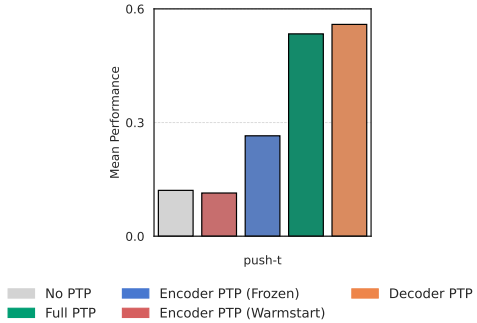

Figure 4: Effect of past-token prediction.

3.1 Past-Token Prediction

Past-token prediction (PTP) has been implicitly used in prior works on policy learning Chi et al. (2023b); Lee et al. (2024b), yet its impact has not been systematically analyzed or ablated. In this work, we seek to provide a detailed characterization of PTP and its effect on policy learning.

As illustrated in Figure 3, past-token prediction involves training a policy to reconstruct both past actions $a'_1, \ldots, a'_{T-1}$ and future actions $a'_T, \ldots, a'_{T+C}$ from an observation sequence. The used loss function is applied over all predicted actions to match target past and future actions:

$$\sum_{a'_t \in \pi(o)} loss(a'_t, a_t) \quad for \quad t \in [1, T+C]$$

As we show in Section 4, this auxiliary objective plays a crucial role in enabling policies to leverage long histories effectively. Notably, we find that PTP matches or even outperforms chunking-based methods when combined with transformer decoders. Furthermore, PTP enables us to train policies that learn to solve tasks that without history even with chunking cannot be solved.

3.2 Cache Embeddings from a Pretrained Encoder

While PTP improves policy robustness, training long-horizon policies remains computationally expensive due to the need to process large observation histories in GPU memory. To the benefit of the method, we learn that if we use PTP we can use a frozen encoder trained on no-history context (which is faster to train than a long-history encoder) and use that one frozen to only train the decoder of the long-history policy, as we show in 4. Thereafter, this insight that we can decouple the training of the observation encoder training it with short history, from the action decoder allow us to address this challenge.

Formally, consider a policy π^H composed of $\pi^H_{enc}(o_i) = \phi_i$, which encodes each observation step (of one or more images and proprioception) and $\pi^H_{dec}(\phi_1, \ldots, \phi_H) = \{a_1, \ldots, a_n\}$, which decodes actions from these embeddings. On the test environment of a transformer-based Diffusion Policy, we first fit π^2 for 500 epochs. We then initialize $\pi^H_{enc} = \pi^2_{enc}$ for the best checkpoint we've found and cache embeddings $D' = \{\pi^2_{enc}(o_i), o_i \forall o_i \in D\}$, training π^H_{dec} for another 500 epochs using the embeddings in D'.

Since these embeddings $\pi_{enc}(o)$ have dimensionality two orders of magnitude smaller than that of the raw images, the training efficiency increases significantly (Section 3.2). Intuitively, at history length h in t epochs each observation in the dataset D is processed ht times for a total cost of $O(|D|ht)$. This cost comes to dominate training time for long history-conditioned policies. By caching the embeddings and computing them only at the beginning of training, we can reduce this encoding cost to $O(|D|)$, obtaining a significant increase in computational efficiency.

3.3 Inference-Time Verification

Beyond training efficiency and regularization, we bring the insight that we can leverage PTP at test time to enhance policy robustness by introducing a self-verification mechanism. This mechanism allows the policy to select the most reliable action sequence by measuring consistency with its past predictions.

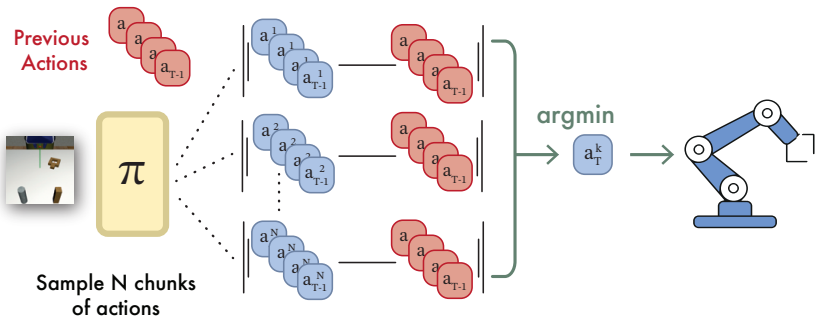

Figure 5: Test-time verification. Multiple action sequences are sampled from the same observation, and the policy selects the sequence that is most consistent compared to ground-truth previous actions.

Given an observation sequence o, we sample N candidate action sequences:

$$\mathcal{A} = \{A_1, \ldots, A_N\}, \quad \text{where} \quad A_i \sim \pi(o) \tag{1}$$

Each sampled sequence $A_i = \{a_1, \ldots, a_{T+C}\}$ contains actions for both past $(a_1, \ldots, a_{T-1})$ and future timesteps. Since the first $T-1$ actions have already been executed, we use them as a reference to evaluate each sampled sequence.

To select the most reliable sequence, we compute a similarity metric between predicted and actual past actions. Intuitively, if a sampled sequence accurately reconstructs the past, it is more likely to generate high-quality future actions. We define the similarity metric as the L2 distance:

$$\mathrm{d}(a, a') = \sum_{a_t, a'_t \in a, a'} ||a_t - a'_t||^2 \tag{2}$$

Thus, we select the optimal action sequence A^* as:

$$A^* = \text{argmin}_k \quad \mathrm{d}(a_k, a_{\text{past}}), \quad \text{for} \quad a_k \in \mathcal{A} \tag{3}$$

As illustrated in Figure 5, this method helps keep the policy in-distribution and improves performance, particularly on long-horizon tasks. Notably, inference-time verification enables policies trained for shorter histories or fewer epochs to achieve performance comparable to fully trained models, providing a simple way of balancing between training efficiency and deployment reliability.

4 Experiments

In this section, we empirically evaluate whether our proposed method allows us to efficiently train strong history-conditioned policies. We focus on the following questions:

1. Is Past Token Prediction (PTP) crucial for policies to improve with increasing context? What does it affect?
2. Does PTP match low-history performance on current benchmarks and solve tasks requiring long history?
3. Can PTP maintain strong performance with closed-loop control?
4. Does embedding caching maintain policy performance compared to end-to-end PTP training?
5. How much can caching embeddings speed up training?
6. Can we use inference-time self-verification to improve policy performance without further training?

Of these questions, the first two relate to *understanding* the model, the next two to *evaluation*, and the last two *efficiency*. We present the results in this order.

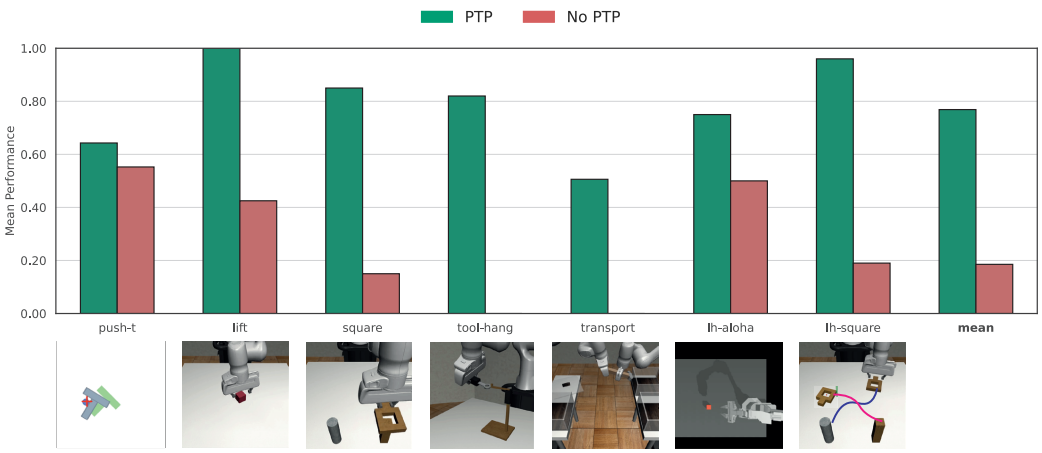

Figure 6: Comparison of 16-observation policy performance with and without PTP on simulation tasks. PTP leads to improvement on all policies, and is especially crucial for realistic, complex tasks and tasks requiring history.

4.1 Environments and Datasets

In order to answer the questions mentioned above, we set up evaluations on robomimic (Mandlekar et al., 2021), a set of simulation environments extensively used by the community. In robomimic, a policy manipulates a set of objects with a Franka Emika arm using visual and proprioceptive observations. We train on multi-human datasets as multimodality may increase the need for history, evaluating on the *lift*, *square*, *tool hang*, and *transport* tasks. Additionally, we use the PushT environment introduced in Diffusion Policy (Chi et al., 2023b), as past experiments have demonstrated a need for action consistency there.

In addition to existing benchmarks, which generally require only limited policy memory, we also set up two simulated tasks that require long history in order to be solved. First, we gather a *long-horizon* square dataset based on the square task, where the robot needs to place the square on the furthest peg from its initial position. This requires substantial history to remember the initial position throughout a rollout. We train on a set of 100 scripted demonstrations collected with substantial noise in order to make it impossible to tell from position or direction alone which peg was the farthest from the start. We also propose a second long-horizon task using the simulated ALOHA robot, where one of the arms needs to pick a block, move it to the center of the field of view, and place it at the position where it was originally, which also requires long history.

4.2 Analyzing Past Token Prediction

We first study the effects of Past Token Prediction. Previous models have used Past Token Prediction implicitly without ablating or justifying its use, so we begin by training Diffusion Policy and VQ-BeT (Lee et al., 2024b) at various history lengths with and without Past Action Prediction. We find on PushT that both DP-Transformer and VQ-BeT performance collapses with increasing context length if past action prediction is not enabled, while performance increases with history if past action prediction is enabled (Fig. 12). Interestingly, past token prediction does not allow DP-CNN to cope with history: we theorize history-conditioned models must be able to attend to a large receptive field in the history for good performance.

We then evaluate DP-Transformer with and without PTP on a wide suite of tasks (6). In all cases, we find significant boosts in performance using PTP, especially on the more visually-complex robomimic tasks. In long-horizon square, a task designed to need history, we measure differences of 70% in success rate, while on the complex Tool Hang and Transport tasks the policy has zero success without Past Token Prediction.

Next, we study two possible hypotheses of the mechanism of PTP's effect. One possibility is that PTP encourages nuisance information to be removed from the visual encoder, as previous works have tried to do explicitly (Seo et al., 2023; Wen et al., 2020a). Another option is that PTP encourages robust representations in the decoder, after merging history steps together.

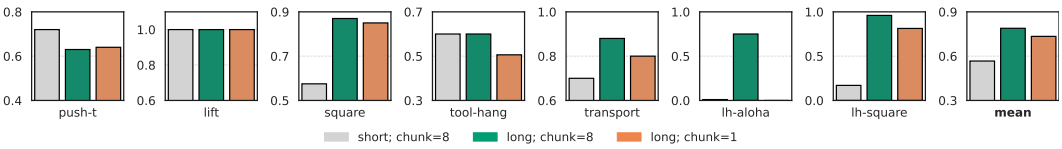

Figure 7: We compare long-horizon policies to baseline low-history results with action chunking. History-conditioned policies usually match or improve on short-horizon performance while maintaining strong results when executed closed-loop. Note ALOHA is not executed without action chunking due to its differing frame rate, making comparison difficult.

To study if PTP regularizes the encoder, we take an encoder from a policy trained with PTP on the PushT task and load it into a randomly-initialized decoder during training. We evaluate two variants: "warmstart," where the encoder is allowed to change after loading, and "freeze," where the encoder is frozen for the duration of training. To evaluate whether PTP instead regularizes the decoder, we load an encoder from a low-history policy (which shouldn't be able to remove any nuisance correlates) and freeze the encoder for the duration of training. We find that only using PTP on the decoder is sufficient for the policy to perform well, while the same is not true for the encoder (see Figure 4). The results also suggest that features from a low-history policy are sufficient for good performance with long contexts.

This discovery that PTP acts on the decoder suggests that our policy can learn from history regardless of our encoder, so long as the encoded representation is rich enough to accurately predict current and future actions. The frozen short-context encoder mentioned previously has the benefit of naturally encoding such rich features, since even a low-history policy can predict future actions accurately based on the features it provides.

We further test this Decoder PTP protocol on Lift, and find that as in PushT we match the full PTP performance. Due to useful efficiency boosts that we can obtain from caching (4.4), we adopt Decoder PTP as our default implementation for the wider suite of robomimic tasks.

4.3 Comparison to Short-Context

Many state-of-the-art robot learning policies avoid history for the reasons mentioned previously. Instead, they use *action chunking* (Zhao et al., 2023a), where multiple actions are predicted and executed from a single timestep without replanning. This approach is appealing because it frees the model from needing to learn how to predict temporally-consistent actions, resulting in strong performance without using substantial history.

We consider how PTP compares to these short-context policies on our set of benchmarks. We use the low-history configurations provided with DP-Transformer to train a policy, and compare it to a PTP policy with context length 16. We see the history-conditioned policy usually matches or outperforms the short-history policy when executing action chunks of size 8 (Figure 7). We see an especially high boost (almost 80%) in performance in our long-horizon task, where open-loop action execution is not sufficient to preserve the history information needed. This is a strong positive signal that our policy has indeed learned to make use of history information beyond action chunking.

Evaluating policies at various context lengths on closed-loop control serves to emphasize the increased performance of history-conditioned policies (compare 2 and 16 observations in Figure 8 or see Table 1 in the appendix). Policies with short context lengths must rely on action chunking, as they are unable to use policy history to identify previous strategies. Thus, on several tasks, the closed-loop performance of short history tasks drops to near zero.

On these same tasks, our policy maintains strong performance. This independence from action chunking suggests our policy has learned to capture relevant strategy information from its action history, allowing it to maintain action consistency and stay in distribution. The short policy must rely on action chunking to fill this role, which may limit reactivity and adaptability.

As a further test, we also evaluate our PTP policies at various context lengths on closed-loop control, expecting to see performance improvements as history length increases. This is indeed the case, as shown in Figure 8. We note that in settings like PushT the effect size is smaller, perhaps due to the relative simplicity of the inputs, while the change is greatest in complex tasks like transport or tasks requiring substantial history, like our long-horizon square task.

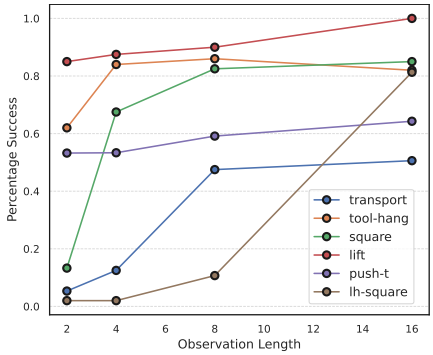

Figure 8: Comparision of robot policies with various observation lengths. Our method enables the learned policy to consistently benefit from longer historical contexts.

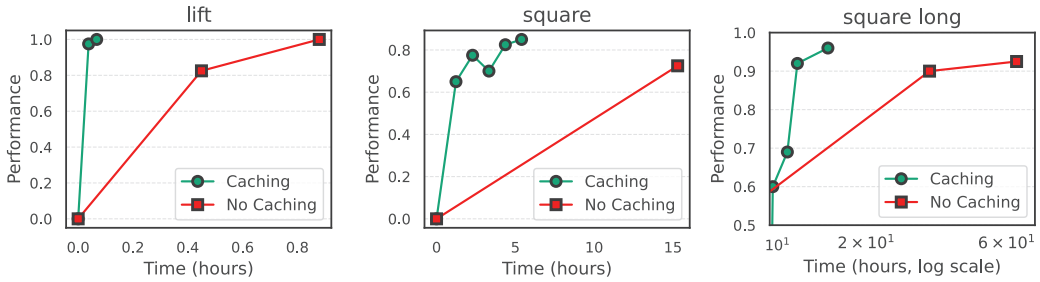

Figure 9: Comparison of PAP policies trained with caching versus those trained with a frozen encoder. Caching significantly enhances performance across various training durations, as it allows the policy to train for substantially more epochs within a fixed time budget.

4.4 Quantifying the Efficiency of Caching

The previous insights on PTP for decoding allow substantial speedups during training, as cached embeddings can reduce observation encoding time to a constant. Further, due to the decrease in overall input size from the embeddings, GPU memory access time as a proportion of overall training time also decreases. To quantify our training efficiency, we compare training times for our Full PTP and Decoder PTP methods on Lift, Square, and long-horizon Square (as all other tasks take prohibitively long to train without caching). We plot performance against compute cost and find a substantial speed-up at every point on the performance curve (Figure 9). We find that for a given amount of training time, we're able to substantially improve results with encoder caching as it enables us to train for many more epochs.

Note that the training time of the cached embeddings policy already takes into consideration the time to pretrain the encoder which is still the main bottleneck in this process.

4.5 Inference-Time Verification

We additionally probe whether the past token outputs of our model can be used to improve the performance of policies (Figure 10). Our hypothesis is that sampling several times and choosing the sample with greatest consistency with past actions will help the policy stay closer to the distribution. We test this on both Square and Lift, using checkpoints trained either with substantially fewer training epochs or checkpoints overfit with truncated history. We see significant improvements in both settings, with one policy even slightly exceeding the level of our best single-sample policy.

4.6 Real-World Experiments

We finally evaluate wether the key observations from our simulation experiments hold in real-world environments. We set up a task where a robotic arm must pick up a block from one side of a table and correctly recall its pickup location to place it on the opposite side. As shown in Figure 11, a baseline policy without history completes the task only 40% of the time. Qualitatively, we observe

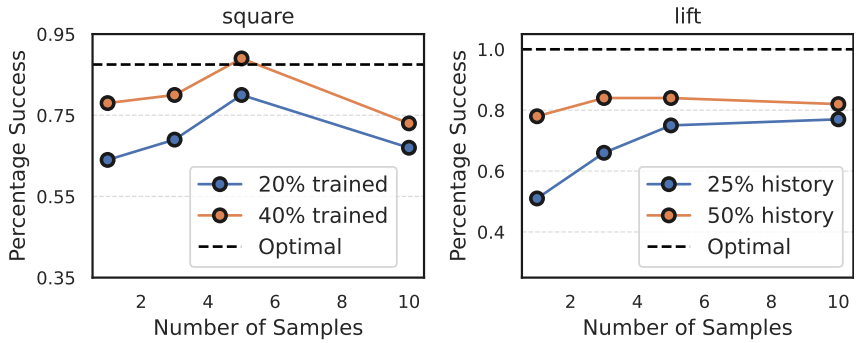

Figure 10: We evaluate inference-time verification on two different settings, and show that taking more samples may improve performance of policies trained for shorter periods of time or overfit to shorter history lengths to near the performance of optimally-tuned policies for that task.

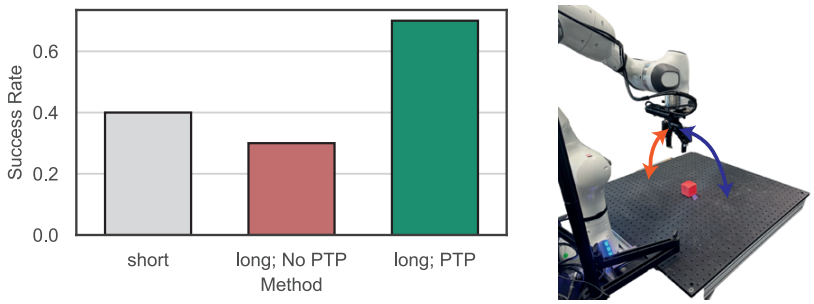

Figure 11: Comparsion of different policies on the real-world long-horizon task. The long-context policy trained by our method achieves ∼2x higher success rate compared to existing approaches.

that after picking up the block, the arm often moves it to an arbitrary side instead of the correct opposite location, as it lacks memory of where the block was originally picked up. In contrast, a policy trained with historical context correctly retains the pickup location and consistently places the block in the correct spot. Moreover, we find that past token prediction (PTP) is essential for training long-context policies—without it, performance deteriorates, even compared to policies that do not use history at all.

5 Discussions

Summary. In this work, we characterize a previously unstudied auxiliary objective in robot learning, past token prediction, as crucial for history-conditioned policy performance. Using our insights, we present an improved recipe for efficient training of effective long-context policies. Our contributions are three-fold:

1. *Understanding Past Token Prediction.* Previous literature gives conflicting views on whether history can be used in policies without feature engineering. We show that predicting the past action is a natural regularizer that allows our policy decoder to improve with history even given input with many spurious features.
2. *Efficient Training of Long-Context Policies.* Motivated by our observation that Past Token Prediction acts on the decoder of our policy, we cache visual embeddings of our policy, leading to an extreme speedup in training of history-conditioned policies, especially for long image input sequences. This lets us train stronger history policies $10\times$ faster than naive training.
3. *Inference-Time Self-Verification.* We show that PTP can be used at inference-time to select samples more likely to stay in distribution, improving suboptimal policies substantially. This idea of self-verification has previously been well-explored in language modeling: PTP-trained policies allow us to use these principles to improve robot learning.

Limitations and Future Work. While this paper conducts a detailed study of the effects of past token prediction on long-context policies, there are several areas for future development. First, while

we demonstrate that PTP is essential for policy improvement with history on multiple imitation learning policies and robots, our optimizations primarily focus on the single-task setting. As robot policies scale, a natural extension is to see if these methods improve performance on VLAs. Some prior work has already found promise for history in these policies (Li et al., 2024b), but no work has explored using past action prediction or a larger cached encoder (e.g., a vision language model). Further, the principles underlying PTP can inform more complex objectives to improve effective decoding of actions from an observation history with many spurious correlations. Additionally, we explore inference-time verification as a method to enable tradeoffs between training time, history length, and inference time when training history-conditioned policies. Our verification strategy is simple, and future work could potentially explore more subtle objectives.

REFERENCES

Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, May 2009.

Hritik Bansal, Arian Hosseini, Rishabh Agarwal, Vinh Q. Tran, and Mehran Kazemi. Smaller, Weaker, Yet Better: Training LLM Reasoners via Compute-Optimal Sampling, August 2024.

Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. RoboAgent: Generalization and Efficiency in Robot Manipulation via Semantic Augmentations and Action Chunking. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4788–4795, May 2024.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2024. URL https://arxiv.org/abs/2410.24164.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Robotics: Science and Systems XIX*. Robotics: Science and Systems Foundation, July 2023a. ISBN 978-0-9923747-9-2.

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)*, 2023b.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems, November 2021.

Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning, 2019. URL https://arxiv.org/abs/1905.11979.

Siddhant Haldar, Zhuoran Peng, and Lerrel Pinto. BAKU: An Efficient Transformer for Multi-Task Policy Learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, November 2024.

Seungjae Lee, Yibin Wang, Haritheja Etukuru, H. Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior Generation with Latent Actions, March 2024a.

Seungjae Lee, Yibin Wang, Haritheja Etukuru, H. Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior generation with latent actions. *arXiv preprint arXiv:2403.03181*, 2024b.

Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards Generalist Robot Policies: What Matters in Building Vision-Language-Action Models, December 2024a.

Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards generalist robot policies: What matters in building vision-language-action models, 2024b. URL https://arxiv.org/abs/2412.14058.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. In *The Twelfth International Conference on Learning Representations*, October 2023.

Yuejiang Liu, Jubayer Ibn Hamid, Annie Xie, Yoonho Lee, Maximilian Du, and Chelsea Finn. Bidirectional Decoding: Improving Action Chunking via Closed-Loop Resampling, December 2024.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Inference-Time Scaling for Diffusion Models beyond Scaling Denoising Steps, January 2025.

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, 2021. URL https://arxiv.org/abs/2108.03298.

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 *Proceedings of the 5th Conference on Robot Learning*, pp. 1678–1690. PMLR, January 2022.

Mitsuhiko Nakamoto, Oier Mees, Aviral Kumar, and Sergey Levine. Steering Your Generalists: Improving Robotic Foundation Models via Value Guidance. In *8th Annual Conference on Robot Learning*, September 2024.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-Scale Simulation of Everyday Tasks for Generalist Robots.

Soroush Nasiriany, Sean Kirmani, Tianli Ding, Laura Smith, Yuke Zhu, Danny Driess, Dorsa Sadigh, and Ted Xiao. Rt-affordance: Affordances are versatile intermediate representations for robot manipulation, 2024. URL https://arxiv.org/abs/2411.02704.

Harish Ravichandar, Athanasios S. Polydoros, Sonia Chernova, and Aude Billard. Recent Advances in Robot Learning from Demonstration. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(Volume 3, 2020):297–330, May 2020.

Juntao Ren, Priya Sundaresan, Dorsa Sadigh, Sanjiban Choudhury, and Jeannette Bohg. Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning, 2025. URL https://arxiv.org/abs/2501.06994.

Stephane Ross and Drew Bagnell. Efficient Reductions for Imitation Learning. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 661–668. JMLR Workshop and Conference Proceedings, March 2010.

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning, 2011. URL https://arxiv.org/abs/1011.0686.

Seokin Seo, HyeongJoo Hwang, Hongseok Yang, and Kee-Eung Kim. Regularized behavior cloning for blocking the leakage of past action information. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 2128–2153. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/06b71ad997f7e3e4b2e2f2ea12e5a759-Paper-Conference.pdf.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. URL https://arxiv.org/abs/2408.03314.

Jonathan Spencer, Sanjiban Choudhury, Arun Venkatraman, Brian Ziebart, and J. Andrew Bagnell. Feedback in imitation learning: The three regimes of covariate shift, 2021. URL https://arxiv.org/abs/2102.02872.

Kaya Stechly, Karthik Valmeekam, and Subbarao Kambhampati. On the Self-Verification Limitations of Large Language Models on Reasoning and Planning Tasks, August 2024.

Priya Sundaresan, Quan Vuong, Jiayuan Gu, Peng Xu, Ted Xiao, Sean Kirmani, Tianhe Yu, Michael Stark, Ajinkya Jain, Karol Hausman, Dorsa Sadigh, Jeannette Bohg, and Stefan Schaal. Rt-sketch: Goal-conditioned imitation learning from hand-drawn sketches, 2024. URL https://arxiv.org/abs/2403.02709.

Dian Wang, Stephen Hart, David Surovik, Tarik Kelestemur, Haojie Huang, Haibo Zhao, Mark Yeatman, Jiuguang Wang, Robin Walters, and Robert Platt. Equivariant Diffusion Policy. In *8th Annual Conference on Robot Learning*, September 2024.

Chuan Wen, Jierui Lin, Trevor Darrell, Dinesh Jayaraman, and Yang Gao. Fighting copycat agents in behavioral cloning from observation histories, 2020a. URL https://arxiv.org/abs/2010.14876.

Chuan Wen, Jierui Lin, Trevor Darrell, Dinesh Jayaraman, and Yang Gao. Fighting Copycat Agents in Behavioral Cloning from Observation Histories. In *Advances in Neural Information Processing Systems*, volume 33, pp. 2564–2575. Curran Associates, Inc., 2020b.

Chuan Wen, Jierui Lin, Jianing Qian, Yang Gao, and Dinesh Jayaraman. Keyframe-focused visual imitation learning, 2021. URL https://arxiv.org/abs/2106.06452.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. Large language models are better reasoners with self-verification, 2023a. URL https://arxiv.org/abs/2212.09561.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. Large Language Models are Better Reasoners with Self-Verification. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, December 2023b.

Fei Yu, Anningzhe Gao, and Benyou Wang. OVM, Outcome-supervised Value Models for Planning in Mathematical Reasoning. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 858–875, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

Maryam Zare, Parham M. Kebria, Abbas Khosravi, and Saeid Nahavandi. A Survey of Imitation Learning: Algorithms, Recent Developments, and Challenges. *IEEE Transactions on Cybernetics*, 54(12):7173–7186, December 2024.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3D Diffusion Policy, March 2024.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023a. URL https://arxiv.org/abs/2304.13705.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware, April 2023b.

Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies, 2024. URL https://arxiv.org/abs/2412.10345.

6 APPENDIX

6.1 PTP ON VARIOUS ARCHITECTURES

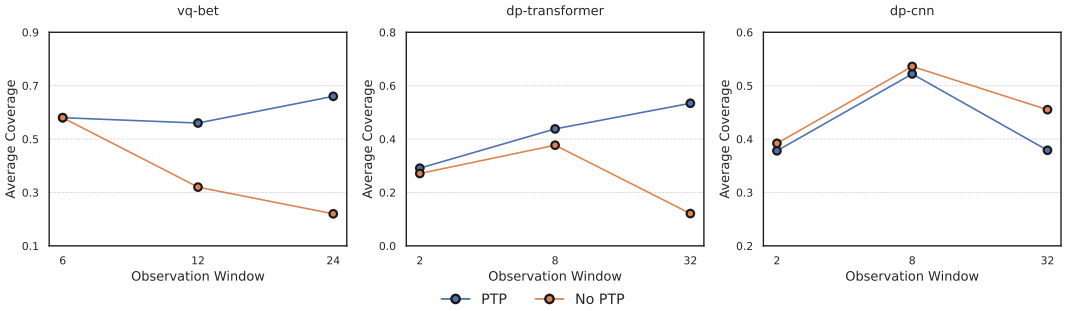

Figure 12: Model performance at various context lengths (on a log scale) with and without past token prediction

We evaluate DP-Transformer, DP-CNN, and VQ-BeT on Image PushT with and without past token prediction. For DP-CNN, performance decreases in long observation windows both with and without PTP, while in both DP-T and VQ-BeT there is a striking deviation between no PTP, which performs worse at extreme time horizons, and PTP, which improves further at extreme time horizons (12).

We hypothesize that transformer-based architectures perform better than the CNN-based architecture due to the large effective receptive field of the transformer, which can attend to all history steps, compared to the CNN, which implicitly biases timesteps to pay more attention to neighboring timesteps. This may suggest that for history-conditioned policies to be successful, the policy must be able to model history interactions over long timeframes (like the transformer). For this reason, we proceed over the rest of the paper with evaluations on a transformer-based architecture.

6.2 AGGREGATED RESULTS FOR EACH COMPONENT

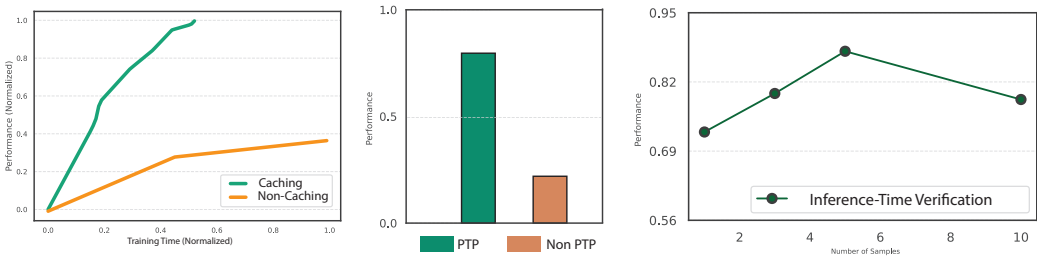

Figure 13: Our method trains a non-history policy to produce a pretrained encoder for useful frame embeddings, and then fine-tunes a history decoder with past token prediction (PTP). This approach yields substantial benefits on policy performance and training efficiency, which can be augmented with inference-time verification by validating past token prediction.

6.3 RAW SUCCESS RATES

We list the raw success rates reported for short and long-horizon policies with and without chunking. History policies are all trained without chunking, and then the best of their top two checkpoints evaluated with chunking is reported as the top result. Baseline (two-observation) policies are trained with chunking, and the same protocol applied to their top checkpoints. We further report the mean over all tasks for success rates without chunking to highlight the trend of improving performance with larger context lengths.

Table 1: Success rate (%) on various robomimic tasks, compared to baseline checkpoint with 2 observation steps, chunk size=1

Observations	Transport	Tool Hang	Square	Lift	Push-T	Long Square	Mean
2	0.053	0.62	0.13	0.85	0.53	0.02	0.37
4	0.13	0.84	0.68	0.88	0.53	0.02	0.51
8	0.48	**0.86**	0.83	0.9	0.59	0.11	0.63
16	**0.51**	0.82	**0.85**	**1**	**0.64**	**0.81**	**0.77**

Table 2: Success rate (%) on various robomimic tasks, compared to baseline 2-observation, chunk size=8

Observations	Lift	Square	Transport	Tool Hang	Long Square	ALOHA	PushT
2	**1**	0.575	**0.6**	0.7	0.04	0	**0.72**
16	**1**	**0.87**	**0.6**	**0.88**	**0.96**	**0.75**	0.63