

CroSTata: Cross-State Transition Attention Transformer for Robotic Manipulation

Giovanni Minelli¹, Giulio Turrisi¹, Victor Barasuol¹, Claudio Semini¹

{giovanni.minelli, giulio.turrisi, victor.barasuol, claudio.semini}@iit.it

Abstract—Learning robotic manipulation policies through supervised learning from demonstrations remains challenging when policies encounter execution variations not explicitly covered during training. While incorporating historical context through attention mechanisms can improve robustness, standard approaches process all past states in a sequence without explicitly modeling the temporal structure that demonstrations may include, such as failure and recovery patterns. We propose a Cross-State Transition Attention Transformer that employs a novel State Transition Attention (STA) mechanism to modulate standard attention weights based on learned state evolution patterns, enabling policies to better adapt their behavior based on execution history. Our approach combines this structured attention with temporal masking during training, where visual information is randomly removed from recent timesteps to encourage temporal reasoning from historical context. Evaluation in simulation shows that STA consistently outperforms standard cross-attention and temporal modeling approaches like TCN and LSTM networks across all tasks, achieving more than 2× improvement over cross-attention on precision-critical tasks. The source code and data can be accessed at the following link: <https://github.com/iit-DLSLab/croSTata>

I. INTRODUCTION

Imitation learning (IL) has emerged as a promising paradigm for training robotic policies by leveraging expert demonstrations rather than learning a policy from scratch through extensive interaction with the environment [1]. The appeal of IL lies in its data efficiency and ability to leverage human expertise, making it particularly attractive for complex manipulation tasks [2], [3], [4], [5], [6]. However, a fundamental limitation of IL approaches lies in the inherent dependence on the statistical distribution of training data, leading to brittle policies that struggle to handle situations not explicitly observed during training [1], [7], [8]. This becomes even more relevant when deploying these models in unstructured and real-world scenarios where environmental conditions, object properties, or execution dynamics may differ from those observed in demonstrations [7], [9], [10].

To address this distributional shift problem, recent work has explored using suboptimal or noisy demonstration data, showing that sufficient diversity can sometimes outperform expert-only training, especially for long-horizon tasks [11], [12]. This has motivated data augmentation approaches and automated demonstration generation systems [4], [13], [14], [15], [16], [17], [18] as methods to include diversity and failure/recovery trajectories in training data, thereby providing explicit examples of mistakes and corrections.

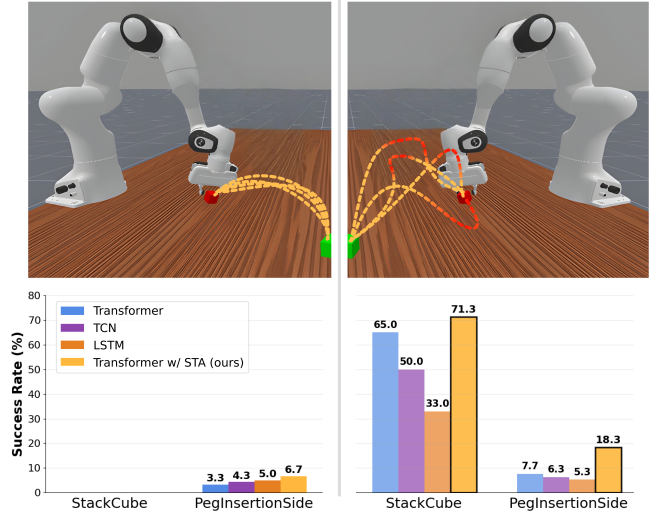


Fig. 1: Performance comparison on simulated manipulation tasks when training with successful-only demonstrations (left) versus recovery-rich demonstrations (right). Our State Transition Attention (STA) mechanism shows particular effectiveness at exploiting temporal patterns in recovery-rich data, achieving superior performance compared to standard temporal modeling approaches.

However, while enriching demonstrations with failure and recovery examples provides a valuable training signal, simply adding more demonstrations is not a scalable solution. In general, it would be extremely difficult to collect examples for every possible failure scenario. This fundamental limitation highlights the need for approaches that can better leverage the underlying causal dependencies present in the data beyond straightforward sequence imitation. These dependencies span multiple aspects: logical dependencies where low-level actions depend on high-level plans; spatial dependencies where end-effector orientation depends on position; and crucially, temporal dependencies where future actions depend on past execution history. Alternative approaches based on planning [19] and hierarchical [20] policy architectures address some of these challenges with promising results, but the fundamental question of how to extract dependency concepts from data and effectively model them in a policy remains central to achieving robust and adaptive behavior.

The temporal modeling challenge is particularly relevant

¹Istituto Italiano di Tecnologia, Genoa, Italy

because many robotic tasks are inherently non-Markovian: appropriate action selection often depends not only on current observations, but also on past observations and actions [5], [21], [22]. For example, manipulation scenarios where the robotic arm occludes critical scene information [23], or multi-stage tasks where early execution steps inform later strategies [3], [24]. In these cases, information from past actions and decisions – such as speed, trajectory curvature, or overall strategy – fundamentally determines the execution of future actions. Despite the importance of historical context, learning long-context robotic policies through imitation learning remains challenging due to spurious correlations in extended observation histories [25].

Current sequence modeling approaches in robotics predominantly treat all temporal elements equally, learning relationships between past and present states primarily through the statistical co-occurrence of elements in demonstrated trajectories [26]. While this approach has shown success in various domains, it may not optimally exploit the structured temporal dependencies in rich demonstrations, where specific past states directly inform corrective actions; thus, more targeted attention mechanisms could better capture these state transition relationships.

We propose a state transition attention mechanism that shifts attention-based temporal processing to focus specifically on how the past informs current action selection. Rather than extracting information from past timesteps and learning how to weight attention across the temporal dimension, our approach directly learns to act based on state transition patterns. This allows policies to leverage historical context by matching current situations to learned temporal patterns during action selection. We evaluate our approach against standard attention mechanism for temporal modeling and demonstrate its particular effectiveness in learning from recovery-rich demonstrations. Moreover, through analysis and ablations of the proposed method, we provide insights into how historical information is retrieved during execution phases, demonstrating that our structured attention mechanisms designed for state transition modeling can significantly enhance policy robustness in sequential decision-making.

The main contributions of the paper are:

- State Transition Attention (STA), a novel attention mechanism designed to capture state transition patterns in robotic manipulation that, to the best of our knowledge, is the first approach to explicitly modulate attention weights based on learned state evolution patterns for temporal reasoning in manipulation policies;
- Comprehensive empirical evaluation and ablation studies across four manipulation tasks demonstrating that STA achieves consistent improvements over the standard cross-attention approach (up to 2× on precision-critical tasks) and outperforms traditional temporal modeling methods including TCN and LSTM networks. Our analysis highlights the method’s capability to differentiate between different types of historical context during policy execution.

II. RELATED WORK

A. Temporal Modeling and Attention Mechanisms

The importance of historical context in robotic tasks has motivated various temporal modeling approaches. Classical methods employ Temporal Convolutional Networks (TCNs) [27] and Long Short-Term Memory (LSTMs) [28] modules for sequence processing, while more sophisticated approaches like dynamic neural advection models incorporate temporal skip connections to handle occlusion through context conditioning on previous observations [29]. The L-MAP framework addresses temporal sequence modeling by learning temporally extended macro-actions for scalable decision-making [30]. Recent transformer-based [31] approaches rely on sequential modeling capabilities for temporal decision-making in robotics. Decision transformers and trajectory transformers apply sequence modeling within reinforcement learning frameworks, modeling task execution as autoregressive sequence prediction [32], [33]. In contrast, imitation learning approaches employ sequence modeling through different strategies: methods like VIMA demonstrate capabilities for long-horizon tasks and generalization [4], while others predict sequences of actions in chunks [3], [22], [34], [35]. In-context learning approaches leverage Transformers’ sequential nature by enriching input sequences with example trajectories that attention mechanisms can retrieve from [36]. Standard attention mechanisms have proven excellent for sequence modeling applications, demonstrating strong noise resistance and long-sequence information retrieval capabilities [37]. Nonetheless, they primarily learn state relationships through statistical co-occurrence rather than explicitly modeling the implications of state evolution. This limitation can lead to failures when execution variations or world state interpretations diverge from training distributions. Recent work has begun addressing this limitation through bidirectional approaches that enforce coherence between past and future predictions [25], [24], yet the fundamental challenge of effectively leveraging structured temporal dependencies in recovery-rich demonstrations remains largely unexplored.

III. METHOD

Effective utilization of historical information becomes critical in robotic manipulation tasks where current observations may be insufficient due to occlusion, ambiguous scene configurations, or partial observability. To achieve this, we need both a mechanism capable of modeling how past states relate to current conditions and training data with informative temporal patterns that can aid learning. Our preliminary tests demonstrate that recovery-rich demonstrations—which contain explicit failure-to-recovery patterns—consistently improve performance across different temporal modeling methods compared to training on successful trajectories alone (Fig. 1). This suggests that such demonstrations provide the necessary informative temporal structures for learning robust policies. Building on this insight, we propose a state transition attention mechanism designed specifically to exploit these temporal patterns.

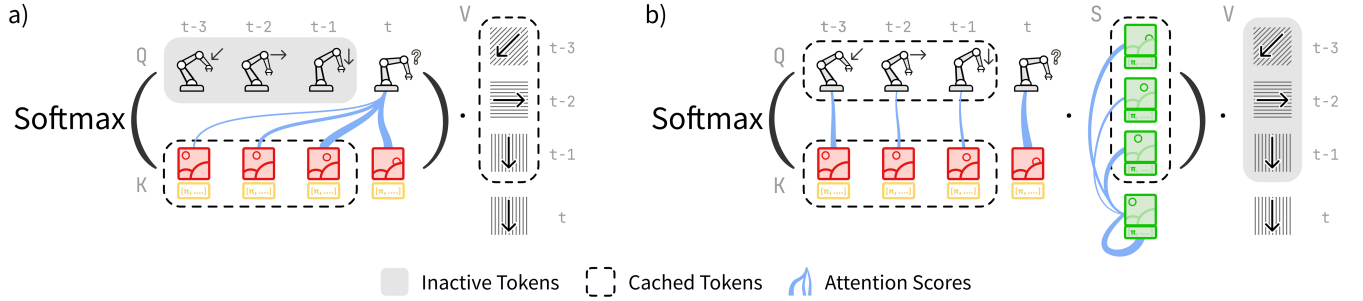


Fig. 2: Graphical representation of how cross-attention works within the Transformer architecture adopted. Here, the query (Q) tokens represent joint values, while the keys (K), values (V) and state (S) tokens encode the overall system information. On the left, a) illustrates standard cross-attention and, on the right, b) depicts State Transition Attention.

A. State Transition Attention Mechanism

We propose a novel modification to standard attention mechanisms that focuses on state transition patterns rather than individual past states. Our approach is motivated by the intuition that the relational patterns, most relevant for current decision-making, emerge from understanding how states evolve over time, particularly, observing direct relationships between subsequent states.

Following an architectural approach similar to [35], we adopt an encoder-decoder structure employing cross-attention to relate action representations in the decoder to state information from the encoder. For example, in the case of robotics, this could be relating planned joint movements (actions) to visual observations and sensor readings (states). Standard cross-attention mechanisms in this setup learn to relate current actions to all present and past states through learned linear projections. These projections must learn representations valid across all temporal distances, while positional embeddings distinguish different timesteps and softmax operations weight the relevance of distant past events. This approach places a significant burden on the attention mechanism to both learn appropriate representations and correctly dampen irrelevant historical information through softmax normalization.

Instead, we shift the computational focus toward interpreting state transition patterns, by using relationships between current and past states to reproject attention scores relative to the different timesteps (i.e., between past action tokens and past state tokens). In (1), we formalize standard cross-attention applied to a temporal sequence, and in (2) we formalize our proposed State Transition Attention (STA) mechanism:

$$\text{Softmax}\left(\frac{Q_t K_{t-k:t}^T}{\sqrt{d_K}}\right) V_{t-k:t} \quad (1)$$

$$\text{Softmax}\left(\frac{\text{diag}(Q_{t-k:t} K_{t-k:t}^T)(S_{t-k:t} S_t^T)}{\sqrt{d_K d_S k}}\right) V_t \quad (2)$$

where Q, K, V, S represent independent linear neural network projections of decoder (Q) and encoder (K, V, S) tokens, subscripts indicate the elements of the sequence from time t back to k elements in the past, and finally,

d_K and d_S refer to the dimension of the relative variable projection. The state transition projection S learns to identify which historical states are most relevant given the current state, creating transition-aware attention values that are then multiplied with the diagonal elements of the QK^T matrix, i.e., $Q_{t-k} K_{t-k}^T, \dots, Q_t K_t^T$.

This operation introduces an additional projection matrix but maintains computational efficiency at inference through standard caching strategies [38] for Q, K and S while restricting the expensive softmax computation to include only recent timesteps rather than the entire sequence. Additionally, since attention scores between Q and K are no longer processed based on temporal position, we incorporate positional embeddings directly into the transition projection output, allowing the model to maintain temporal awareness while focusing on state evolution patterns. A graphical comparison between attention strategies is provided in Fig. 2.

B. Architecture Design

Building upon the STA component, we design our decoder with standard Transformer blocks, each comprising cross-attention (implemented with STA) and self-attention modules followed by a feed-forward network. This decoder processes *input tokens* representing robot actions that encompass coordinated motion across all joints. To better exploit relational patterns through self-attention mechanisms, we use multiple *input tokens* per single robot action, initialized with proprioceptive information and combined with absolute positional embeddings [31] to establish intra-relationships between them. After being processed by the Transformer blocks, the output tokens are individually processed through a multi-layer perceptron (MLP) to produce the target action for each joint, then executed through an underlying PD controller. We refer to this decoder architecture as STA Transformer for brevity in the remainder of the paper.

The encoder deals with world state information processing through convolutional neural networks (CNN) and MLP modules that handle visual and proprioceptive inputs, respectively. The concatenated output of these modules represents the *state tokens* for the decoder's cross-attention component. This architecture allows the model to process the evolution of the world state (through cross-attention with STA)

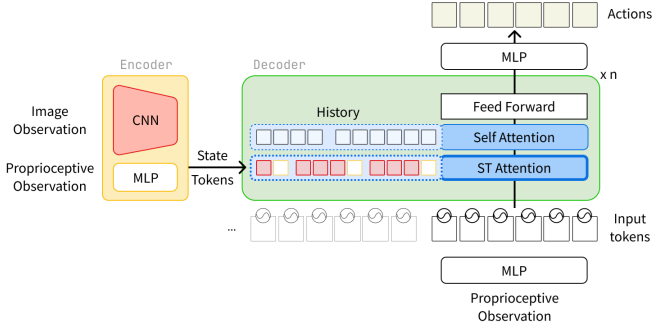


Fig. 3: Architecture overview of our proposed Transformer with STA. The encoder processes visual observations through CNN and proprioceptive data through MLP to generate *state tokens*. The decoder employs standard self-attention for *input token* interactions (white squares) and our novel STA module as cross-attention with current and historical *state tokens* (colored squares). Both decoder input’s tokens and encoder state-related tokens are being cached for reuse in later steps.

and to model the robot’s internal kinematics (through self-attention), providing both stages with tokens of current and past timesteps. A complete overview of the architecture is provided in Fig. 3.

C. Training Strategy with Temporal Masking

For training, we sample sequences of 16 timesteps where each one uses the previous steps as contextual history, while future information remains masked. We optimize the model using mean squared error (MSE) loss between predicted and ground-truth actions. Additionally, to incentivize information retrieval from historical context and enhance learning of temporal dependencies, we propose a temporal masking strategy applied to visual inputs provided to the encoder. This approach removes all exteroceptive information available at current timesteps, forcing the model to rely on historical context for decision-making. We apply this masking to k consecutive timesteps in each sampled sequence (excluding the first/oldest timestep), where k is randomly sampled from $[2, L/2]$ with L being the total sequence length. This masking strategy serves a dual purpose: it prevents the model from developing over-reliance on current visual information while simultaneously encouraging the development of robust temporal reasoning capabilities over information available in the history.

IV. EVALUATION

A. Evaluation Setup

To evaluate the effectiveness of our STA mechanism, we design experiments using simulated manipulation tasks where policies can potentially benefit from historical context during execution. We evaluate our method on four tasks from the ManiSkill benchmark [14] selected for their distinct failure modes: *StackCube* (Fig. 4a) and *PegInsertionSide* (Fig. 4b) require precise manipulation; *TwoRobotStackCube* (Fig. 4c) demands coordinated manipulation of two objects

simultaneously where failure in either subtask compromises the overall task; and *UnitreeGITransportBox* (Fig. 4d) requires complex coordination involving synchronized torso rotation and arm movements. These tasks feature short horizons (less than 5 seconds) but remain failure-prone due to their demanding execution requirements, making them ideal testbeds for evaluating temporal reasoning capabilities under randomly generated conditions at inference time. Each task uses one or two cameras attached to the workspace table and the robotic manipulator to provide visual information, while proprioceptive state information consists exclusively of joint position values. Action prediction consists in joint position delta values across all evaluated methods. Additional environment details are provided in Fig. 4.

B. Data Collection

We collect successful demonstrations containing artificially introduced failure sequences followed by natural recovery behaviors, similar to DAgger methodology [1]. Our data generation process employs policies trained with privileged information using the Proximal Policy Optimization (PPO) algorithm. During trajectory collection, we artificially introduce failure patterns by sampling noise at random execution moments to modify the state representation of world elements for n consecutive steps, forcing the policy to perform suboptimal actions (e.g., misperceiving object positions leading to failed grasps or collisions). Trained to be noise-robust, these policies naturally exhibit recovery behaviours towards the real target after the expiration of the n steps, thus creating demonstrations that contain explicit failure-correction patterns. Crucially, while we artificially induce failures, we label the collected trajectory steps to train exclusively on noise-free action predictions. Although predictions for failure-induced steps do not contribute to the training loss, the visited states and generated trajectory segments remain key components of the sequential network input, providing rich temporal context for learning state transition patterns. We generated 1000 episodes per task for training.

C. Baseline Methods

We compare our STA Transformer against four baseline approaches: a Transformer that processes temporal sequences with standard cross-attention, one that does not take past historical context into account, and established temporal modeling approaches including TCN [27] and LSTM [28] networks. All networks present relatively small architectures with 4 processing layers and a hidden size of 512. We equipped TCN and LSTM networks with additional final feed-forward networks to predilige a fair performance comparison given the parameter count differences with transformer-based methods. The encoder structures remain comparable across all methods, with a hidden representation of 512. Additional implementation details and training hyperparameters are provided in Table I.

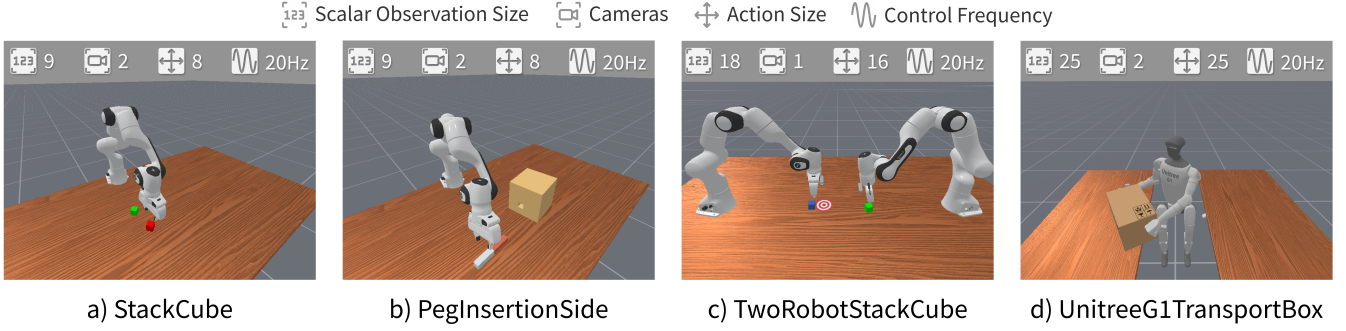


Fig. 4: ManiSkill manipulation tasks used for evaluation. (a) *StackCube*: Single-arm manipulation requiring coordinated grasping and placement; (b) *PegInsertionSide*: Precision insertion task demanding correct orientation and alignment of the peg with the box hole slot; (c) *TwoRobotStackCube*: Bimanual coordination task for collaborative cube stacking in a target location; (d) *UnitreeG1TransportBox*: Multi-joint coordination task involving arm and torso coordination in a humanoid robot to transport a box across the workspace.

D. Performance Evaluation

Fig. 5 presents the performance results in terms of success rate, demonstrating the relative effectiveness of different temporal modeling approaches when learning from recovery-rich demonstrations. Our STA Transformer consistently outperforms all baseline approaches across all four tasks, with particularly notable improvements in precision-requiring tasks such as *PegInsertionSide*, where STA achieves 18.3% success compared to 7.7% for standard Transformer – representing more than a 2x improvement. Interestingly, the standard Transformer approach with full history attention show occasional underperformance compared to no-history baselines in some tasks, suggesting that naive historical information integration does not always benefit policy performance. Additionally, traditional sequence modeling approaches demonstrate notable limitations in this domain, with TCN and LSTM networks consistently showing lower performance. This is particularly true for LSTM, which performs poorly across precision-critical tasks. These results support our hypothesis that superior temporal reasoning capabilities, achieved through our structured attention mechanisms designed for state transition modeling, provide performance advantages compared to both history-agnostic approaches and standard sequence modeling techniques.

TABLE I: Networks architecture details.

	TCN	LSTM	Transformer
Decoder size	512	512	512
Encoder size	512	512	512
N Layers	4	4	4
N Heads	-	-	8
Normalization	LayerNorm	LayerNorm	RMSNorm
FFN Non Linearity	ReLU	ReLU	GELU
Final FFN	✓	✓	✗
Batch Size	16	16	16
Sequence Length	16	16	16
Learning Rate	8e-5	8e-5	8e-5

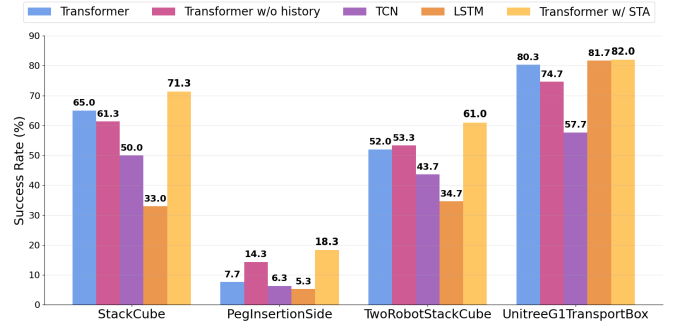


Fig. 5: Success rate comparison across four ManiSkill [14] manipulation tasks. All methods were trained for 50 epochs on recovery-rich demonstrations with periodic validation performed directly in the simulation environment. Results represent the best validation checkpoint performance averaged over 3 seeds with 100 episodes per evaluation. Variance between seeds was negligible and is omitted for clarity.

E. Attention Pattern Analysis

Beyond overall performance metrics, we analyze the learned attention patterns to understand how our proposed mechanism processes temporal information. Specifically, we examine how STA weights historical attention scores, e.g., $Q_{t-k}K_{t-k}^T$ relative to k timestep in the past, through the state transition scores from $S_{t-k}S_t^T$ when situations requiring corrective actions occur. Fig. 6 shows a trajectory executed by our transformer-based policy in the *PegInsertionSide* task, which involves reaching the peg, lifting it from the white end, and inserting it into the box hole. The analyzed scenario demonstrates four key phases: starting phase ($t=4$), failed grasping attempt ($t=10$), successful recovery grasp ($t=29$), and final insertion approach ($t=39$). We examine the patterns of attention modules at these critical moments by observing how the value of state transition scores evolves over time. The visualization shows the values extracted from the last Transformer layer, which we observed to provide the most interpretable results. The top heatmaps display state transition

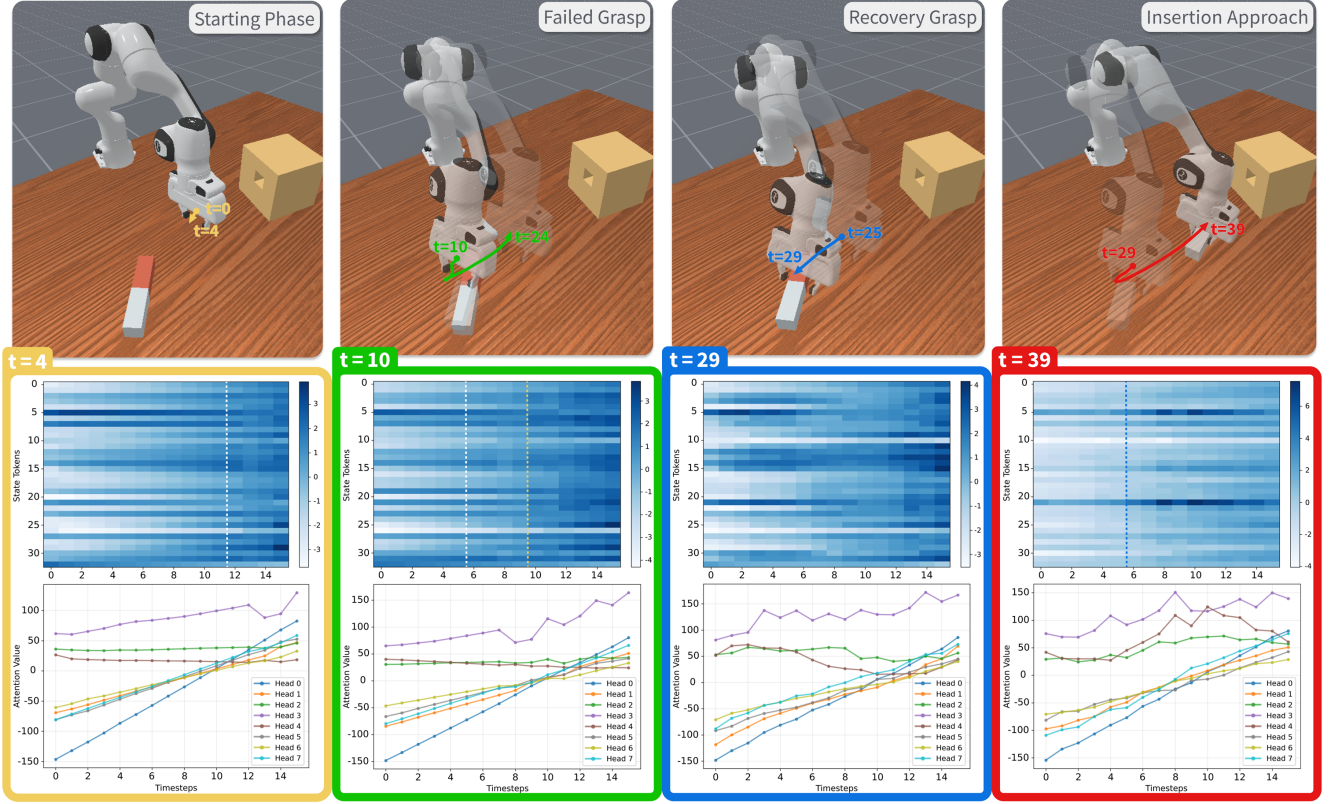


Fig. 6: Attention pattern analysis during a *PegInsertionSide* trajectory showing four critical execution phases. Top heatmaps display attention weights averaged across heads for *state tokens* over a 15-step history window (darker = higher attention). Bottom graphs show per-head attention values summed across tokens. In both visualizations, timestep 0 represents the oldest historical information while timestep 15 represents the present. Vertical dotted lines indicate the corresponding timestep in the temporal sequence for each colored execution phase. The visualization demonstrates how STA learns to retrieve relevant historical context during recovery phases ($t=29$) compared to initial execution patterns ($t=4$, $t=10$).

weights averaged across all heads for *state tokens*, showing their individual relevance across the 15-step history window (darker colors indicate stronger attention). The bottom graphs show state transition values summed across tokens per computation head, revealing which heads contribute the most to historical context retrieval at each timestep. Starting from a pre-initialized history of identical starting states – strategy adopted strictly for analysis purposes – we observe initial patterns showing lower state transition scores for past states and higher scores for current state information ($t=4$). This pattern persists through the first grasping attempt ($t=10$) that ultimately fails. The comparison between the first and second grasping attempts reveals a striking change: during the successful recovery attempt ($t=29$), the patterns show higher state transition scores extending more into the past, with particular activation of heads 2 and 4 that appear to function as retrieval pathways for relevant historical state relationships. The final trajectory phase ($t=39$) shows restored focus on recent timesteps with higher state transition scores on specific tokens, likely facilitating the precise movements necessary for successful peg insertion.

This analysis provides insights into how the model attends to relevant past events in its execution history during

corrective reattempts, while downweighting attention scores in other scenarios, supporting our hypothesis that structured attention mechanisms can better exploit the temporal dependencies present in recovery-rich demonstrations.

F. Impact of Temporal Masking on Training and Inference Robustness

To understand the specific contributions of our temporal masking training strategy, we evaluate both the direct effects of masked training and the robustness it provides at inference time. We compare standard training with masked training under inference conditions with both complete and partially masked observations. Specifically, for ‘masked’ inference evaluation, we remove visual observations for n consecutive steps where n ranges from 0 to $L-1$, with L being the total sequence length. ensuring at least one timestep retains the full observation. The masking duration n is resampled with 0.1 probability when it reaches zero otherwise standard conditions of inference apply. Fig. 7 presents results for the *StackCube* task, revealing several important findings. Our STA Transformer trained with masking achieves superior performance under standard inference conditions compared to the same model trained without masking (71.3% vs 64.7%), demonstrating that the temporal masking strategy

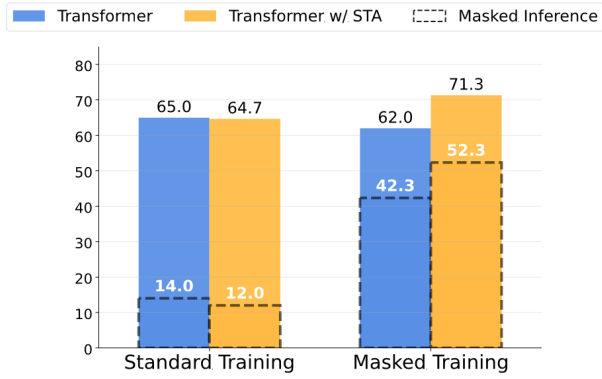


Fig. 7: Ablation study comparing temporal masking effects during training and robustness at inference time. Results show the *StackCube* task performance under different training (standard vs. masked) and inference (complete vs. partially masked observations) conditions. Our proposed Transformer with STA benefits from temporal masking training while standard Transformer shows minimal improvement, demonstrating architecture-specific advantages.

enhances learning even when full observations are available during deployment. This suggests that forcing the model to rely on historical context during training develops more robust temporal reasoning capabilities. In contrast, the standard Transformer baseline does not benefit from masked training under normal inference conditions, with performance remaining essentially unchanged compared to standard training (65.0% vs 62.0%). This indicates that the effectiveness of temporal masking is specifically tied to our state transition attention mechanism rather than being a general improvement applicable to any architecture. Notably, when evaluating robustness under masked inference conditions, our STA Transformer trained with masking strategy maintains a significant advantage over the baseline (52.3% vs 42.3% under masked inference).

G. Historical Context Dependency

We provide additional analysis of how history length affects the performance of our STA Transformer. We focus on *StackCube* and *PegInsertionSide* environments, training our method with 15 historical steps (16-step sequences) and then evaluating performance with variable history lengths. Additionally, we train reference policies on sequences with history lengths of 7, 3, 1, and 0 steps to provide baseline performance where training and inference settings match. Results in Fig. 8 demonstrate the robustness of our approach when evaluated with limited historical information available. For *PegInsertionSide*, we observe only modest performance degradation as inference history decreases, while in *StackCube*, our method occasionally even exceeds the initial performance. This robustness suggests that our STA mechanism enables learning effective decision-making from rich historical information during training, which transfers well to inference scenarios with truncated historical context. Relative to reference policies trained with shorter histories,

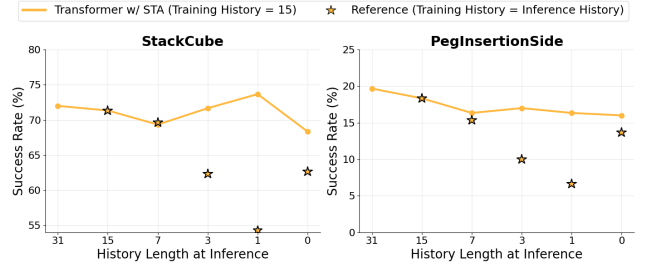


Fig. 8: Historical context dependency analysis showing performance vs. inference history length for Transformers with STA trained with 15-step history. Star markers indicate reference performance from models trained with matching history lengths. Results demonstrate robust performance across different inference history lengths, with *StackCube* showing remarkable stability and *PegInsertionSide* exhibiting graceful degradation.

we observe substantial performance drops, particularly for those trained with 1 and 3 historical steps. We attribute this degradation to our temporal masking procedure during training, which may excessively dampen the training signal when applied to sequences with already limited temporal information. Notably, the reference policy with 0 historical steps at training (where no temporal masking is applied), presents higher performance than those with 1 and 3 historical steps. This interaction between masking strategy and history length highlights the importance of adequate temporal context for our training approach to be effective.

V. DISCUSSION

Our work demonstrates a key insight on how our proposed structured attention mechanisms can effectively learn to exploit temporal dependencies present in recovery-rich demonstrations. This is evidenced by attention pattern analysis showing adaptive retrieval of historical information during recovery phases. However, we acknowledge limitations constraining the generalizability of our findings. The manipulation tasks evaluated are relatively short-horizon and do not strictly necessitate historical reasoning to succeed, as evidenced by the non-trivial performance of baselines omitting historical context. While our improvements target challenging cases where imprecise movements and unrecoverable situations occur, more complex temporally-extended tasks would provide stronger evaluation grounds for historical reasoning capabilities. Additionally, our data collection methodology remains constrained by the capabilities of privileged policies used to generate failure and recovery patterns, limiting the diversity and complexity of temporal dependencies that can be learned.

VI. CONCLUSIONS

We have presented State Transition Attention (STA), a novel attention mechanism that modulates attention weights based on learned state evolution patterns to improve temporal reasoning in robotic manipulation policies. Our experimental evaluation demonstrates that STA consistently

outperforms standard temporal modeling approaches across four ManiSkill manipulation tasks, with particularly notable improvements in precision-critical scenarios. The ablation studies reveal additional findings regarding temporal masking and demonstrate robustness to reduced historical context at inference time. These results establish our approach as a promising direction for developing more capable manipulation policies that can effectively learn from and reason about their execution history. Future work should explore more sophisticated data collection strategies and evaluate performance on complex temporally-extended tasks that more strongly necessitate historical reasoning capabilities.

REFERENCES

- [1] M. Drolet, S. Stepputtis, S. Kailas, A. Jain, J. Peters, S. Schaal, and H. B. Amor, "A comparison of imitation learning algorithms for bimanual manipulation," *IEEE Robotics and Automation Letters*, vol. 9, no. 10, pp. 8579–8586, 2024.
- [2] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 297–330, 2020.
- [3] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, 2024.
- [4] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan, "Vima: General robot manipulation with multimodal prompts," in *International Conference on Machine Learning (ICML)*, 2023.
- [5] S. Lee, Y. Wang, H. Etukuru, H. J. Kim, N. M. M. Shafiullah, and L. Pinto, "Behavior generation with latent actions," in *International Conference on Machine Learning (ICML)*, p. 1076, 2024.
- [6] M. Zare, P. M. Kebria, A. Khosravi, and S. Nahavandi, "A survey of imitation learning: Algorithms, recent developments, and challenges," *IEEE Transactions on Cybernetics*, vol. 54, no. 12, pp. 7173–7186, 2024.
- [7] P. De Haan, D. Jayaraman, and S. Levine, "Causal confusion in imitation learning," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.
- [8] J. Spencer, S. Choudhury, A. Venkatraman, B. Ziebart, and J. A. Bagnell, "Feedback in imitation learning: The three regimes of covariate shift," *arXiv preprint arXiv:2102.02872*, 2021.
- [9] C. Wen, J. Lin, T. Darrell, D. Jayaraman, and Y. Gao, "Fighting copycat agents in behavioral cloning from observation histories," in *Conference on Neural Information Processing Systems (NeurIPS)*, p. 216, 2020.
- [10] O. Mees, L. Hermann, and W. Burgard, "What matters in language conditioned robotic imitation learning over unstructured data," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11205–11212, 2022.
- [11] A. Kumar, J. Hong, A. Singh, and S. Levine, "Should I run offline reinforcement learning or behavioral cloning?," in *International Conference on Learning Representations (ICLR)*, 2022.
- [12] F. Lin, Y. Hu, P. Sheng, C. Wen, J. You, and Y. Gao, "Data scaling laws in imitation learning for robotic manipulation," in *International Conference on Learning Representations (ICLR)*, 2025.
- [13] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3019–3026, 2020.
- [14] T. Mu, Z. Ling, F. Xiang, D. C. Yang, X. Li, S. Tao, Z. Huang, Z. Jia, and H. Su, "Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.
- [15] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox, "Mimicgen: A data generation system for scalable robot learning using human demonstrations," in *Conference of Robot Learning (CoRL)*, 2023.
- [16] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao, X. Yuan, P. Xie, Z. Huang, R. Chen, and H. Su, "Maniskill2: A unified benchmark for generalizable manipulation skills," in *International Conference on Learning Representations (ICLR)*, 2023.
- [17] C. Garrett, A. Mandlekar, B. Wen, and D. Fox, "Skillmimicgen: Automated demonstration generation for efficient skill learning and deployment," in *Conference of Robot Learning (CoRL)*, 2024.
- [18] Z. Zhou, P. Atreya, A. Lee, H. R. Walke, O. Mees, and S. Levine, "Autonomous improvement of instruction following skills via foundation models," in *Proceedings of CoRL'24*, 2024.
- [19] Y. Feng, J. Han, Z. Yang, X. Yue, S. Levine, and J. Luo, "Reflective planning: Vision-language models for multi-stage long-horizon robotic manipulation," *arXiv preprint arXiv:2502.16707*, 2025.
- [20] Y. Li, Y. Deng, J. Zhang, J. Jang, M. Memmel, C. R. Garrett, F. Ramos, D. Fox, A. Li, A. Gupta, and A. Goyal, "HAMSTER: Hierarchical action models for open-world robot manipulation," in *International Conference on Learning Representations (ICLR)*, 2025.
- [21] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," in *Conference on Robot Learning (CoRL)*, pp. 1678–1690, 2022.
- [22] T. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," in *Robotics: Science and Systems (RSS)*, 2023.
- [23] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu, "Robocasa: Large-scale simulation of everyday tasks for generalist robots," in *Robotics: Science and Systems (RSS)*, 2024.
- [24] Y. Liu, J. I. Hamid, A. Xie, Y. Lee, M. Du, and C. Finn, "Bidirectional decoding: Improving action chunking via guided test-time sampling," in *International Conference on Learning Representations (ICLR)*, 2025.
- [25] M. T. Villasevil, A. Tang, Y. Liu, and C. Finn, "Learning long-context robot policies via past-token prediction," in *ICLR'25 Workshop on Robot Learning: Towards Robots with Human-Level Abilities*, 2025.
- [26] M. Wen, R. Lin, H. Wang, Y. Yang, Y. Wen, L. Mai, J. Wang, H. Zhang, and W. Zhang, "Large sequence models for sequential decision-making: a survey," *Frontiers of Computer Science*, vol. 17, no. 6, 2023.
- [27] C. S. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. Hager, "Temporal convolutional networks for action segmentation and detection," in *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 1003–1012, 2016.
- [28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [29] F. Ebert, C. Finn, A. X. Lee, and S. Levine, "Self-supervised visual planning with temporal skip connections," in *Conference of Robot Learning (CoRL)*, vol. 78, pp. 344–356, 2017.
- [30] B. Luo, A. Pettet, A. Laszka, A. Dubey, and A. Mukhopadhyay, "Scalable decision-making in stochastic environments through learned temporal abstraction," in *International Conference on Learning Representations (ICLR)*, 2025.
- [31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Conference on Neural Information Processing Systems (NeurIPS)*, pp. 6000–6010, 2017.
- [32] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: reinforcement learning via sequence modeling," in *Conference on Neural Information Processing Systems (NeurIPS)*, p. 1156, 2021.
- [33] M. Janner, Q. Li, and S. Levine, "Offline reinforcement learning as one big sequence modeling problem," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.
- [34] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, *et al.*, " π_0 : A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.
- [35] NVIDIA, "Gr00t n1: An open foundation model for generalist humanoid robots," 2025.
- [36] L. Fu, H. Huang, G. Datta, L. Y. Chen, W. C.-H. Panitch, F. Liu, H. Li, and K. Goldberg, "In-context imitation learning via next-token prediction," in *NeurIPS'24 Workshop on Open-World Agents*, 2024.
- [37] W. Wang, S. Zhang, Y. Ren, Y. Duan, T. Li, S. Liu, M. Hu, Z. Chen, K. Zhang, L. Lu, X. Zhu, P. Luo, Y. Qiao, J. Dai, W. Shao, and W. Wang, "Needle in a multimodal haystack," in *Conference on Neural Information Processing Systems (NeurIPS)*, p. 649, 2024.
- [38] Hugging Face Team, "Unlocking longer generation with key-value cache," 2025. Accessed: 2025-01-30.