2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

QDM-RNN: Acquisition of High-speed and Robust Behavior from Low-speed Demonstrations

Masaki Yoshikawa[1], Hiroshi Ito[1], Hyogo Hiruma[1,2], and Tetsuya Ogata[1,3]

Abstract— **Imitation learning has gained significant attention as a promising approach to enable flexible and generalizable robot motion generation across diverse tasks. However, existing models often suffer from long inference times, limiting their applicability to high-speed and fine-grained tasks. Moreover, while faster computation enables shorter inference cycles, it introduces new challenges such as overcontrol and motion instability when the inference frequency exceeds the sensor sampling rate. To address these issues, we propose QDM-RNN, a lightweight motion generation model that learns from slow but high-quality demonstrations and remains robust under high-frequency inference. Our method utilizes Softmax Transformation, which discretizes the robot end-effector pose into a high-resolution probability distribution, enabling accurate and smooth trajectory prediction. Furthermore, by incorporating multi-timestep prediction, which simultaneously predicts several future steps, our model mitigates instability arising from the mismatch between sensor and inference rates, ensuring consistent long-horizon motion generation. We validated the effectiveness of our approach through real-world robotic experiments on a Sport Stacking task, which requires both high speed and precision. QDM-RNN maintained high success rates and motion stability at speeds up to seven times faster than demonstrations, and showed robustness to external disturbances including background changes, lighting variations, dynamic object perturbations, and obstacles.**

I. Introduction

Achieving fast and stable execution of diverse motions is one of the critical challenges in deploying robots in real-world environments. Traditionally, high-speed visual recognition in industrial tasks such as assembly and transportation has relied on expensive high-speed cameras and dedicated computing units based on FPGAs (Field Programmable Gate Arrays) [1]. These systems enabled robots to follow predefined, rule-based trajectories, facilitating automated operations in structured environments. However, the set of recognizable objects and environmental conditions was limited, and robots were restricted to fixed trajectories, making them suitable only for fixed factory environments. For tasks requiring real-time perception and motion planning, extensive hand-coded programs were necessary, posing significant development costs for achieving motion diversity.

To overcome these limitations, imitation learning has emerged as a promising approach for robot motion generation [2]. In imitation learning, demonstrations—often provided via teleoperation—are used to collect sensor data that the model learns from, enabling the robot to flexibly respond to environmental changes without explicit programming of each trajectory. For instance, RT-1 [3] successfully executed over 700 language-conditioned tasks by training on more than 130,000 episodes of demonstration data. However, its inference operates at only 3 Hz, making it unsuitable for dexterous and precise actions. Similarly, Diffusion Policy [4], a motion generation method based on diffusion models, learns action distributions from tens to hundreds of demonstrations and generates behavior through a reverse diffusion process, allowing multi-modal behavior. Nonetheless, the reverse process typically requires 20–100 inference steps; increasing step count improves accuracy but incurs significant inference delays. While Denoising Diffusion Implicit Models (DDIM) [5] reduce inference time by an order of magnitude, they suffer from decreased prediction accuracy. These results suggest that while imitation learning expands the range of achievable motions, realizing high-speed motion remains a major challenge.

With the advancement of computational resources, inference cycles can now be significantly shortened. However, this introduces new problems. From a model design perspective, when inference frequency is higher than the sensor sampling rate (e.g., 30 Hz), the same sensor data will be input for multiple sequential predictions. This leads to redundant outputs and increases the risk of overcontrol, where minor sensor noise results in unstable behavior such as unintended oscillations, jerky end-effector motions, or excessive corrective movements due to the system's hypersensitivity [6]. Additionally, in models involving sequential denoising processes, such as diffusion models, excessively short control cycles may result in too little environmental change per step, potentially degrading prediction accuracy [7].

From a data quality perspective, imitation learning is highly dependent on the quality of demonstration data provided by human operators. Teaching robots to perform fast and precise motions is particularly difficult for humans, and even slight deviations or delays during demonstrations can degrade the trained model's final motion quality and stability [8]. Thus, in practice, it is more feasible for human operators to provide slow but stable demonstrations, even when the target execution is intended to be fast.

Given this context, we propose a motion generation model that learns from slow and high-quality demonstration data, yet achieves fast and stable motion prediction at inference time. Specifically, we introduce a Softmax Transformation [9], which enables fine-grained, smooth, and continuous trajectory prediction by discretizing the robot's motion into

[1]All authors are with the Department of Intermedia Art and Science, Waseda University, Tokyo, Japan ma3k1@suou.waseda.jp
[2] Hyogo Hiruma is with the Research & Development Group, Hitachi, Ltd., Tokyo, Japan
[3] Tetsuya Ogata is with the National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan ogata@waseda.jp

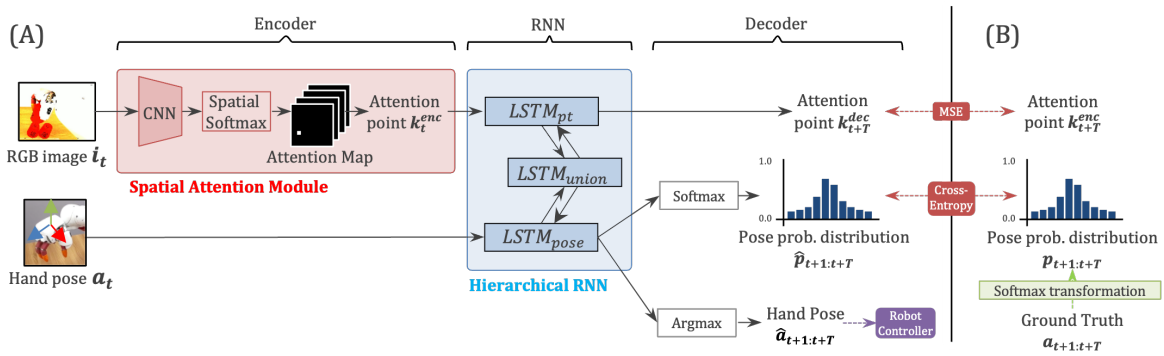

Fig. 1. Overview of the proposed Quantized Distributional Multi-timestep Recurrent Neural Network (QDM-RNN). (A) shows the overall process of the model. The input is the sensor data at timestep t, and the model outputs the attention point at $t+T$ and the hand poses for timesteps $t+1$ through $t+T$. (B) shows the ground truth data, that are processed and used for computing loss for training the model in an end-to-end manner. Note that the hand pose $\hat{a}_{t+1:t+T}$ is only used for executing robot motions and not for training the model.

high-resolution probability distributions. In addition, we implement multi-timestep prediction, allowing consistent and robust motion generation regardless of the control cycle length.

In real-robot experiments, the proposed method successfully performed at seven times the demonstration speed. Moreover, it maintained appropriate motion generation even under environmental disturbances, such as changes in background, variations in lighting, dynamic perturbations to object positions, and the presence of obstacles.

II. Related Work

Imitation learning (IL) for robotic manipulation must address the mismatch that emerges when demonstrations, typically collected at slow and safe velocities, are executed at higher speeds during deployment. Increasing the execution speed significantly alters the physical interactions in the environment, resulting in dynamics that differ from those observed during training. This discrepancy pushes the robot into state regions outside the demonstration distribution, exacerbating prediction errors during execution. Several prior studies have explored different strategies to address this challenge.

The first approach relies on using inherently time-scalable motion representations. Dynamic Movement Primitives (DMPs) encode trajectories in phase space, allowing execution time to be smoothly stretched or compressed by adjusting a single gain parameter while preserving stability [10]. Although elegant, each task typically requires manual tuning of parameters, which quickly becomes fragile and impractical for complex, contact-rich manipulation tasks with many degrees of freedom.

A second approach addresses the issue through data augmentation. Yamamoto et al. artificially accelerated teleoperated grasp trajectories using time warping, enabling their policy to intercept objects moving approximately twice as fast as those seen in the original demonstrations, without requiring additional robot trials [11]. However, purely data-driven augmentation uniformly speeds up all phenomena within the dataset—including physical processes (e.g., free-fall) that should remain consistent—resulting in observations with unrealistic physical interactions. To capture physically accurate high-speed responses, Masuya et al. collected new robot trajectories at varying speeds, by replaying an accelerated version of a slowly taught motion data with the robot, and using them for motion training. This real-world augmentation led to a 55% improvement in wiping and pick-and-place task performance [12]. Despite the performance gains, this approach incurs significant trial costs, as each new velocity setting requires careful supervision and substantial data-collection time. When real-world experimentation is impractical, Continual Domain Randomization (CDR) progressively broadens the range of inertia and friction parameters in simulation, enabling policies to adapt to increasingly faster dynamics [13]. Yet, sim-to-real gaps inevitably widen precisely in nonlinear regimes encountered at high speeds. As a computational alternative, the Proleptic Temporal Ensemble enables robots to execute actions based on ensembles of predictions made at multiple future timesteps. Specifically, by combining predictions computed over several future steps, robots effectively ”skip” inference delays and significantly speed up task execution [14]. However, the computational overhead grows linearly with the look-ahead depth, causing memory and computational bottlenecks on embedded hardware.

A third strategy leverages online expert feedback during model training to mitigate distribution shifts. Dataset Aggregation (DAgger) queries an expert about the learner's own roll-outs, theoretically eliminating compounding errors [15]. SafeDAgger reduces the expert's workload by limiting queries to uncertain states [16], and Deeply AggreVaTeD employs the expert's cost-to-go as supervision, enabling efficient policy-gradient updates even in continuous action spaces [17]. Nevertheless, these methods rely on rapid, millisecond-level expert interventions, which are typically unrealistic in very fast and complex manipulation scenarios.

Despite steady progress, existing methods consistently face trade-offs between human tuning effort, expensive data

collection, run-time computational resources, or expert supervision. Thus, a method that enhances temporal adaptability and enables the model to correctly perceive each unique situation and dynamically adjust its actions on-the-fly remains elusive. The following section introduces our proposed model, which overcomes the aforementioned limitations by leveraging the deep predictive learning framework, along with novel structures that enhance adaptability against temporal variations.

III. Proposed Method

A. Design Concept

The proposed model builds upon the deep predictive learning framework [18], enhanced with additional modules to improve temporal adaptability. Deep predictive learning refers to a framework inspired by the predictive coding theory of human cognition, where the system continuously learns to minimize prediction error for its observed sensorimotor experiences. Deep predictive learning models possess the characteristic ability to achieve high generalization performance from limited data, owing to the attractor dynamics inherent in Recurrent Neural Networks (RNNs) [18].

The model receives sensory inputs consisting of the current visual image i_t and motion information a_t, which includes the 3D pose of the robot arm's end-effector and the gripper state (open/close). It predicts the same modalities as the input but over T future time steps. The training data comprises human teleoperation demonstrations, allowing the model to learn the sensorimotor dynamics associated with the demonstrated actions.

Fig. 1 illustrates the overall architecture of the proposed model, Quantized Distributional Multi-timestep Recurrent Neural Network (QDM-RNN). QDM-RNN consists of three main modules: an Encoder Module, a Recurrent Neural Network (RNN) Module, and a Decoder Module. These modules together project sensory inputs into a latent feature space and enable temporal prediction of future sensory states. The entire model is trained end-to-end (E2E), with all modules jointly optimized to capture consistent dynamics from the demonstration data.

The Encoder Module selectively extracts task-relevant information from multiple visual streams. Based on SA-RNN [19], it applies spatial softmax to CNN-generated feature maps to convert salient image regions—where CNN neurons activate strongly—into 2D coordinates called attention points k_t. This assumes that task-relevant visual regions induce strong neuron activations. By explicitly extracting task-relevant object positions through Spatial Softmax, the proposed model can efficiently capture the information necessary for task execution, even from a limited amount of data.

The RNN Module performs multi-modal temporal prediction using a hierarchical architecture with multiple Long Short-Term Memory (LSTM) units [20]. It consists of separate LSTM streams for visual and motion modalities ($LSTM_{pt}$, $LSTM_{pose}$), along with a unified LSTM ($LSTM_{union}$) that integrates their hidden states. Each modality-specific LSTM predicts its respective future state, and the unified LSTM facilitates information exchange across modalities. This hierarchical setup enables the system to learn fine-grained fast-changing features via modality-specific LSTMs and abstract, slower dynamics through the unified LSTM . Compared to standard LSTMs that directly update internal states at every step, this structure improves robustness to sensor noise and maintains balance between vision and motion learning.

The Decoder Module transforms future predictions from the RNN Module back into modality-specific outputs, namely future attention points and future robot hand poses. These pose predictions are used as control commands for robot actuation. A key novelty lies in this module's new structure, designed to improve temporal adaptability. This is because, simple MLP-based decoders cannot accommodate changes in time scale. In high-frequency inference, where inference exceeds the sensor update rate, inconsistencies arise between predicted actions and sensor readings. Specifically, identical sensory inputs are processed repeatedly despite previous predictions, leading to stagnant or erroneous predictions. To mitigate this, we introduce two mechanisms: Softmax Transformation and Multi-timestep Prediction, detailed in the following subsections.

B. Softmax Transformation

Softmax Transformation converts each continuous-valued component of a vector into a discrete probability distribution over N quantized bins [9]. In our case, the robot state (i.e. 3D pose of robot end-effector and binary state of gripper) is transformed into probability distributions with N discrete values per dimension, as illustrated in Fig. 1(B). Ground truth values of robot pose data are preprocessed into such distribution format. The outputs of $LSTM_{pose}$ are converted into corresponding format with MLPs, along with a softmax function that normalizes to form valid probability distributions.

Conventional models directly regress robot pose using a single neuron per dimension, which lacks representational power and often leads to abrupt, unstable motion. In contrast, we express joint angles as distributions over multiple neurons, resulting in smoother trajectories. Furthermore, the RNN-based $LSTM_{pose}$ retains temporal consistency, enabling smooth and continuous motion prediction over time.

C. Multi-timestep Prediction

The multi-timestep prediction module predicts sensory states not only for the next time step but for up to T future steps. Specifically, $LSTM_{pose}$ generates T-step predictions, and prediction errors are computed using corresponding teacher data across all steps. In contrast to conventional Deep Predictive Learning models [18]–[20], the proposed method explicitly learns and predicts motion while accounting for multi-step temporal dependencies. This approach not only improves training efficiency through batch prediction of multiple steps, but also mitigates the instability arising from mismatches between inference and sensor sampling rates.

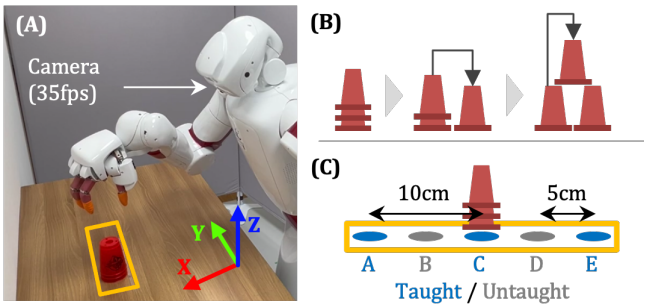

Fig. 2. Experiment setup, where the robot stacks three cups into a pyramid shape. (A) Task environment. (B) The stacking process of the cups. (C) Five different initial positions of the cups, where the data at A, C, E are only given in training data.

Concretely, even when provided with several consecutive identical inputs, the model can infer that they correspond to a previously learned segment of motion rather than an unknown stationary or divergent state, thereby enabling stable prediction.

D. Loss Computation

Loss functions for visual and motion predictions are defined as follows:

$$\begin{cases} Loss = \alpha_{pt} \cdot Loss_{pt} + \alpha_{pose} \cdot Loss_{pose} \\ Loss_{pt} = MSE(k^{dec}_{t+T}, k^{enc}_{t+T}) \\ Loss_{pose} = CrossEntropyLoss(\hat{p}_{t+1:t+T}, p_{t+1:t+T}) \end{cases} \tag{1}$$

Here, MSE denotes mean squared error, and α_{pt}, α_{pose} are weighting coefficients for each modality. The attention point loss $Loss_{pt}$ compares attention predictions from the Decoder and Encoder at matched future time steps. Since ground-truth attention data do not exist, we apply bidirectional consistency loss between Decoder and Encoder outputs, as in [20]. For motion predictions, which are represented as distributions, we use cross-entropy loss to measure divergence from ground-truth distributions. The model is optimized using the Adam optimizer.

IV. Experiments

A. Environmental Setup

Fig. 2(A) illustrates the experimental environment. We employed a humanoid robot equipped with a 7-DoF arm, a 4-finger robotic hand, and a head-mounted camera. The camera is an RGB camera that captures images at 35 frames per second. As task objects, we used plastic cups. For demonstration collection, a game controller was utilized, where the joystick controlled the end-effector position and buttons controlled the hand's open/close state.

B. Task Settings

To evaluate the stability and precision of the proposed model during high-speed execution, we selected the 3-stack task from Sport Stacking, a competitive speed-based activity. This task requires both rapid and accurate motion generation. Fig. 2(B) provides an overview of the task. The robot sequentially picks up three stacked cups from a tabletop, places two cups side by side, and then stacks the third cup on top to form a pyramid shape.

Demonstration data were collected using stacked cups at three different initial positions, arranged linearly with 10 cm spacing. Intermediate positions between these were left out as test cases, to evaluate generalization performance. Task success was defined as forming the correct pyramid structure (two cups side-by-side with one on top) without knocking over any cups. Given the sensitivity of this task to cup placement, precise and stable motion is essential for successful execution, especially at high speeds.

To validate the effectiveness of our method, we generated motions both at slow (10Hz, the same speed as demonstrations) and fast (70 Hz, corresponding to seven times the speed of the demonstrations) command frequencies, and evaluated task success rates at five different positions (A–E). In addition, to evaluate the robustness of the proposed method against environmental changes, we assessed the task success rate when generating motions at 70 Hz with the background color changed to black. Furthermore, we tested the method at 70Hz in various unseen environments, including variations in lighting, dynamic perturbations to object positions, and the presence of obstacles, to comprehensively evaluate its robustness. Here, the term "inference frequency" represents how often the neural network outputs predictions in a second, in contrast to "command frequency" which represents how often the robot receives commands in a second. For instance, if the command frequency is 70 Hz and the multi-timestep prediction horizon is $T = 3$, the resulting inference frequency is approximately $70/3 \approx 23$ Hz. The command frequency of fast motion was set based on the maximum motion speed supported by the robot's hardware.

C. Training Configurations

Demonstration data consisted of a single trajectory per position, resulting in three trajectories in total. Data were sampled at 10 Hz, with each trajectory lasting 45 seconds (450 time steps). Each sample included the robot's end-effector position (2-DoF in the *YZ*-plane as shown in Fig. 2), gripper open/close state (1-DoF), and camera image (650×400 pixels). To reduce GPU memory usage during training, images were resized to 65×40 pixels.

For QDM-RNN, each component of the robot state (three in total) is quantized into $N = 2000$ bins with a temperature parameter of $\sigma = 0.25$, and the model predicts $T = 3$ future steps, resulting in a total of $3 \times 3 \times 2000 = 18{,}000$ output neurons. The values of N, σ, and T were empirically determined through preliminary comparative experiments to achieve a balance between prediction accuracy and computational efficiency. Consequently, during training, the input

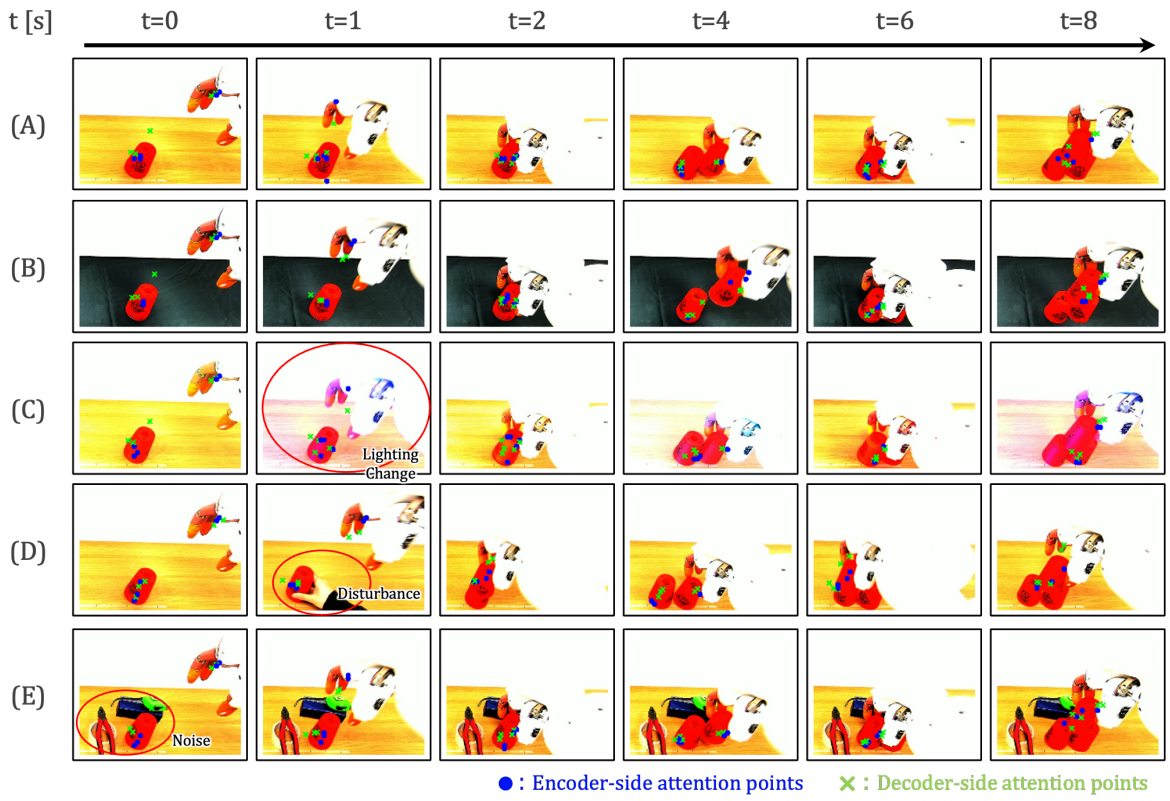

Fig. 3. Prediction results of Sport Stacking task at different conditions. All motions are executed at an untrained speed (seven times faster than the demonstrations). The conditions are as follows: (A) neutral , (B) modified background, (C) modified lighting, (D) disturbance by experimenter during motion, (E) addition of visual noises.

TABLE I

SUCCESS RATE OF THE SPORT STACKING TASK [%]

Pos.	QDM-RNN			ACT			ACT w/o TE		Ablation Studies		
	Slow (10Hz)	Fast (70Hz)	Black BG	Slow (10Hz)	Fast (70Hz)	Black BG	Slow (10Hz)	Fast (70Hz)	w/o Hierarchical RNN	w/o Softmax Transformation	w/o multi-timestep Prediction
A	90	90	90	60	70	0	10	0	70	100	0
B	90	100	70	100	80	0	0	0	80	70	0
C	100	90	100	80	90	0	30	10	30	10	0
D	100	90	100	50	70	0	40	10	90	100	0
E	100	100	100	60	50	0	0	20	40	90	0
Ave.	96	94	92	70	72	0	16	8	62	74	0

sequence length provided to the model was 150 steps for both motor information and camera images (derived from $\frac{450}{3}$), while all 450 steps of motor information were utilized for loss computation. The number of attention points to be learned was set to 5, and the dimensionalities of the hidden states in the modality-specific and unified LSTM modules were set to 50 and 20, respectively.

The total loss function comprised prediction errors for both attention point positions and end-effector positions. Modality-specific losses were weighted by $\alpha_{pt} = 0.1$ for attention point prediction and $\alpha_{pose} = 1.0$ for pose prediction. The model was trained for 10,000 epochs, and the weights achieving the lowest prediction error on the test data were selected for motion generation.

V. RESULTS AND DISCUSSION

A. Task Performance

We first evaluated the performance of the proposed model. Table I presents task success rates (10 trials per condition), and Fig. 3 shows image sequences captured from the robot's onboard camera during inference.

As shown in Table I, QDM-RNN achieved success rates near or above 90% at both trained and untrained positions, regardless of inference speed or under the condition where the background was changed to black, indicating stable and robust task execution. Fig. 3(A) displays the robot's motion and predicted attention points under high-speed inference, showing accurate perception of the objects and the robot's hand, as well as correct and sequential manipulation. Fig. 3(B)–(E) illustrate inference results under various unseen

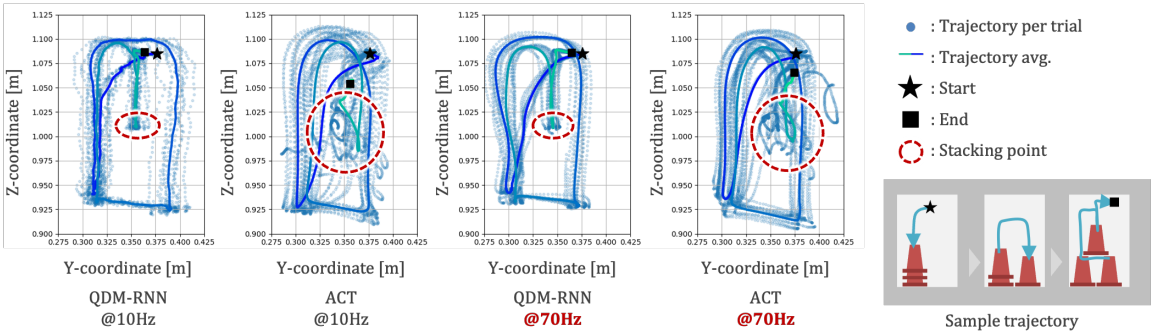

Fig. 4. Visualization of robot-hand's trajectory at untaught position D within Y-Z coordinate space. Each graph displays the result of different model type ("QDM-RNN" (proposed) and "ACT") and command frequency (10Hz and 70Hz). The illustration on the bottom right shows the expected trajectory of the hand in the Y-Z coordinate space.

environmental conditions: (B) background change, (C) lighting change, (D) manual perturbation of object positions by a human operator, and (E) visual noise. In all cases, the attention consistently focused on the target object, and the desired actions were successfully executed.

In contrast, the success rates for the comparison model Action Chunking with Transformers (ACT) [2], one of the representative imitation learning models, are shown in Table I. For ACT, we conducted experiments under two settings: with and without Temporal Ensembling (TE) [2]. For clarity in notation, we refer to the model with Temporal Ensembling as ACT, and the one without it as ACT w/o TE.

Regardless of inference speed, ACT achieved lower precision and an overall success rate around 70%. This suggests difficulty in achieving stable performance at both demonstrated and novel positions. Frequent failure cases included misalignment of the stacked cup (in height or lateral position), causing structural inconsistencies and failed stacking. These errors likely stem from unstable predictions of the end-effector position used to place the cups (see Fig. 4). Moreover, ACT performed no successful motions under any of the unseen environments, revealing ACT's critically low robustness.

ACT predicts multiple future steps at once through action chunking, and computes the final prediction as a weighted average of these predictions via TE [2]. Therefore, by removing TE, it becomes possible to reduce the inference frequency below the sensor frequency during fast motion generation, similar to our proposed method. To evaluate the performance under such a condition, we conducted experiments using ACT w/o TE. As shown in Table I, this model exhibited significantly lower success rates than ACT at all experimental positions. A key characteristic of its behavior was the discontinuity observed at the boundaries of action chunks. Without TE, the model performs inference for a fixed temporal window and generates actions blindly based on that output. When executing motions faster than during training, hardware and environmental discrepancies can cause drift, especially toward the end of the prediction window. Consequently, the actual state may deviate from the predicted one, requiring the model to correct for accumulated errors in the next inference. This correction introduces abrupt trajectory changes at every inference, ultimately degrading task success rates. Therefore, TE is essential for generating smooth motions in ACT. Based on this finding, the following discussions consider only the original ACT as the baseline.

B. Stability Comparison on Modified Temporal Dynamics

To examine robustness to changes in temporal dynamics, we compared the variability in end-effector trajectories generated under different inference rates. Fig. 4 shows the transition of end-effector positions generated by QDM-RNN and the baseline ACT model during slow and fast motion generation at untaught position D. QDM-RNN consistently produced similar motions across speeds and trials, where the visualized hand pose trajectories had low variance from the averaged path. This consistency was especially evident when stacking the third cup, where precise placement is critical—resulting in convergent and stable predictions.

Fig. 5 visualizes the internal state trajectories of the unified LSTM across time steps using the first two principal components. These trajectories were plotted for different starting cup positions. During slow motion (Fig. 5(A)), all trials begin from the same internal state and diverge into distinct trajectories corresponding to each position (A–E), reflecting accurate spatial encoding. Notably, the internal state transitions mirrored the actual spatial layout, confirming that spatial information was correctly embedded in the model state.

Fig. 5(B) shows that during fast motion, QDM-RNN exhibited similar internal state transitions as in the 10 Hz case, demonstrating prediction consistency even under accelerated conditions. Furthermore, Fig. 5(C) displays the states when the cup was manually moved from position C to A during execution (as in Fig. 3(D)). The internal state initially followed the trajectory for C but then smoothly transitioned to the trajectory for A after the intervention, indicating that the unified LSTM adapted seamlessly in real time—even under high-frequency inference.

On the other hand, while ACT's average trajectory appeared similar to QDM-RNN, the variance across trials was significantly larger. This lack of stability, even during slow

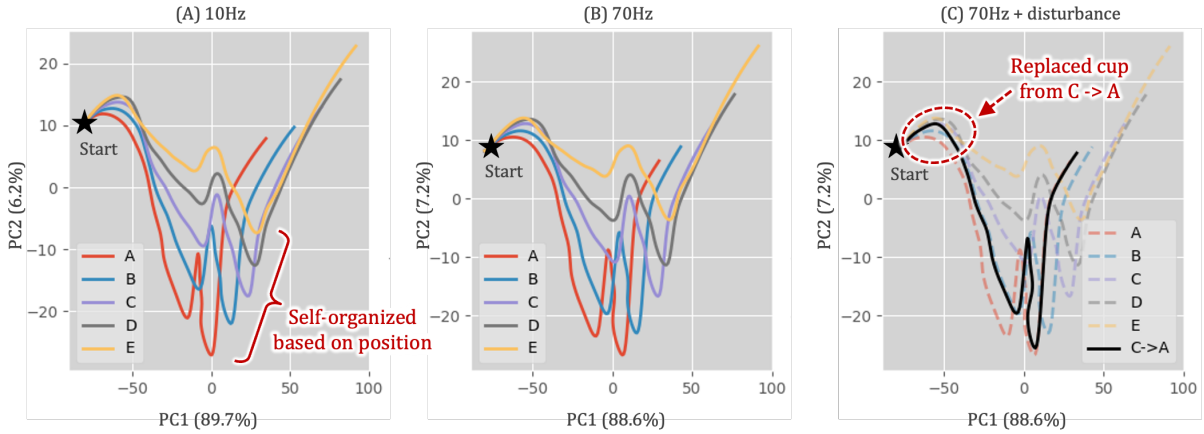

Fig. 5. Transition of hidden states of the unified LSTM during motion generation at different conditions. The hidden states are compressed with principal component analysis (PCA) to two dimensions. (A) Prediction results at 10Hz, (B) prediction results at 70Hz, and (C) prediction results at 70Hz but with disturbance during motion.

TABLE II

MODEL PARAMETERS AND FLOPS

Model	Parameters	Resolution [pixel]	FLOPs
ACT	83.9M	60 x 45	7.3G
		600 x 450	39.2G
Diffusion Policy	91.2M	60 x 45	62.4G
		600 x 450	102G
QDM-RNN	1.27M	60 x 45	32.4M
		600 x 450	3.5G

TABLE III

TASK EXECUTION TIME IN HIGH-SPEED MODE [S]

Model	A	B	C	D	E	Ave.
ACT	10.0	9.8	9.4	9.1	8.8	9.4
QDM-RNN	9.1	9.0	8.6	8.0	7.8	8.5

motion, led to inconsistent motion generation. In particular, large trial-to-trial variation occurred during the final stacking phase, resulting in reduced task success.

C. Ablation Studies

To assess the contribution of the Hierarchical RNN, Softmax Transformation, and multi-timestep prediction modules introduced in the proposed method, we conducted an ablation study evaluating task success rates during motion generation at 70Hz command frequency. Each ablated model was trained using the same dataset as the full QDM-RNN. As shown in Table I, all ablated configurations yielded lower success rates than the full model. Below, we detail the behavioral characteristics observed for each variant:

1) QDM-RNN without Hierarchical RNN: Although the overall precision remained relatively high, the success rate at each position declined compared to the full model. Qualitatively, the robot's end-effector exhibited oscillatory behavior during motion. This instability likely stems from the removal of the hierarchical structure, which impairs the model's ability to capture long-term dependencies. These findings suggest that the hierarchical structure plays a crucial role in stabilizing motion prediction under high-frequency inference.

2) QDM-RNN without Softmax Transformation: This configuration exhibited significant difference in success rates across positions, to as low as 10% at position C. Especially, the motion during relocation of the picked cup (i.e., the sliding up, right, then down to the next position) became inaccurate shortcutting with a curved trajectory, instead of passing through checkpoints as in the demonstration data. Although this smoothed and continuous trajectory occasionally led to higher success rates at specific positions—surpassing even our proposed method due to its stability, it frequently caused the grasped cup to collide with neighboring cups, ultimately resulting in task failure. These failures appear to result from inaccurate interpolation due to the conventional approach of predicting each pose value with a single neuron, which is not preferred under requirement of precise motions. In contrast, the model equipped with Softmax Transformation suppressed such undesired interpolations and helped to faithfully reproduce learned motion patterns even under high-speed inference.

3) QDM-RNN without multi-timestep prediction: This model failed to complete the task at all initial positions. The generated motions were noticeably slower, discontinuous, and frequently resulted in early release of the cup which is before reaching the correct stacking position. These failures likely stem from two factors: redundant inferences due to the inference frequency exceeding the camera frame rate, and the inability to learn dependencies across future timesteps. These results indicate that multi-timestep prediction not only alleviates bottlenecks caused by limited sensor update rates but also contributes to acquiring smooth and continuous motion dynamics.

D. Comparison on Computational Cost

Table II summarizes the number of model parameters and floating-point operations (FLOPs) for QDM-RNN, ACT, and Diffusion Policy. To ensure fair comparison across models

with different input resolutions, we report FLOPs for two settings: 60 × 45 pixels and 600 × 450 pixels. QDM-RNN has approximately 1.27 million parameters, and requires only 32.4M FLOPs at 60 × 45 resolution—indicating extremely lightweight computation. In contrast, ACT and Diffusion Policy both exceeded 80M parameters. Notably, even at the lower resolution, Diffusion Policy requires 62.4G FLOPs, making it computationally expensive. At higher input resolution (600 × 450), ACT and Diffusion Policy reached 39.2G and 102G FLOPs, respectively. Meanwhile, QDM-RNN's FLOPs only slightly increased to 3.5G, demonstrating superior scalability and efficiency. These results show that QDM-RNN's computational cost scales favorably with input resolution, enabling fast inference even with high-resolution inputs. This efficiency makes the method particularly suitable for real-time robotic applications.

Table III reports average task execution times under high-speed conditions. QDM-RNN completed tasks faster than ACT. We attribute this difference to ACT's inference frequency (70 Hz) exceeding the camera frame rate (35 fps), leading to redundant predictions based on identical visual inputs. This mismatch likely caused execution delays. In contrast, QDM-RNN's architecture predicts up to 3 future steps at once via multi-timestep prediction, enabling cross-frame planning and decoupling inference from sensor update rates. As a result, QDM-RNN achieves both fast and stable motion generation. These findings suggest that appropriate tuning of inference frequency with respect to the sensor sampling rate is critical for achieving both rapid and robust robot behavior.

VI. Conclusion

In this study, we proposed QDM-RNN, a lightweight motion generation model that combines Softmax Transformation and multi-timestep prediction to achieve stable and accurate motion generation even with high-frequency command prediction exceeding the sensor sampling rate. Through real-world robotic experiments on a Sport Stacking task, we demonstrated that the proposed method enables precise end-effector control and achieves both high-speed and stable execution. Notably, QDM-RNN maintained high motion stability and task success rates even at a command prediction of 70 Hz, which is seven times faster than the demonstration rate.

On the other hand, the tasks addressed in this study involved relatively simple hand motions constrained to horizontal and vertical planar movement. As future work, we aim to extend the applicability of our method to more complex manipulation tasks involving 3D spatial motion and to scenarios requiring whole-body coordinated control, thereby further expanding the versatility and generalizability of the proposed framework.

Acknowledgments

This work was supported by JST [Moonshot R&D] Grant-Number [JPMJMS2031].

References

[1] T. Senoo, A. Namiki, and M. Ishikawa, "High-speed batting using a multi-jointed manipulator," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 2. IEEE, 2004, pp. 1191–1196.

[2] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.

[3] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[4] 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*, p. 02783649241273668, 2023.

[5] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[6] M. Monforte, L. Gava, M. Iacono, A. Glover, and C. Bartolozzi, "Fast trajectory end-point prediction with event cameras for reactive robot control," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4036–4044.

[7] S. Park, J. Kim, and G. Kim, "Time discretization-invariant safe action repetition for policy gradient methods," *Advances in Neural Information Processing Systems*, vol. 34, pp. 267–279, 2021.

[8] Y. Yuan, X. Li, Y. Heng, L. Zhang, and M. Wang, "Good better best: Self-motivated imitation learning for noisy demonstrations," *arXiv preprint arXiv:2310.15815*, 2023.

[9] A. Ahmadi and J. Tani, "How can a recurrent neurodynamic predictive coding model cope with fluctuation in temporal patterns? robotic experiments on imitative interaction," *Neural Networks*, vol. 92, pp. 3–16, 2017.

[10] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 2. IEEE, 2002, pp. 1398–1403.

[11] K. Yamamoto, H. Ito, H. Ichiwara, H. Mori, and T. Ogata, "Real-time motion generation and data augmentation for grasping moving objects with dynamic speed and position changes," in *2024 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2024, pp. 390–397.

[12] N. Masuya, H. Sato, K. Yamane, T. Kusume, S. Sakaino, and T. Tsuji, "Variable-speed teaching-playback as real-world data augmentation for imitation learning," *arXiv preprint arXiv:2412.03252*, 2024.

[13] J. Josifovski, S. Auddy, M. Malmir, J. Piater, A. Knoll, and N. Navarro-Guerrero, "Continual domain randomization," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 4965–4972.

[14] H. Park, D. Lim, S. Kim, and S. Park, "Proleptic temporal ensemble for improving the speed of robot tasks generated by imitation learning," *arXiv preprint arXiv:2410.16981*, 2024.

[15] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[16] J. Zhang and K. Cho, "Query-efficient imitation learning for end-to-end autonomous driving," *arXiv preprint arXiv:1605.06450*, 2016.

[17] W. Sun, A. Venkatraman, G. J. Gordon, B. Boots, and J. A. Bagnell, "Deeply aggrevated: Differentiable imitation learning for sequential prediction," in *International conference on machine learning*. PMLR, 2017, pp. 3309–3318.

[18] H. Ito, K. Yamamoto, H. Mori, and T. Ogata, "Efficient multitask learning with an embodied predictive model for door opening and entry with whole-body control," *Science Robotics*, vol. 7, no. 65, p. eaax8177, 2022.

[19] H. Ichiwara, H. Ito, K. Yamamoto, H. Mori, and T. Ogata, "Spatial attention point network for deep-learning-based robust autonomous robot motion generation," *arXiv preprint arXiv:2103.01598*, 2021.

[20] H. Hiruma, H. Ito, H. Mori, and T. Ogata, "Deep active visual attention for real-time robot motion generation: Emergence of tool-body assimilation and adaptive tool-use," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8550–8557, 2022.