2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Domain Randomization-free Sim-to-Real : An Attention-Augmented Memory Approach for Robotic Tasks

Jia Qu[1], Shun Otsubo[1],Tomoya Yamanokuchi[2], Takamitsu Matsubara[2] and Shotaro Miwa[1]

***Abstract*— The sim-to-real gap, a long-standing challenge in the field of robotics, has garnered significant attention. Essentially, it is important to learn robust representation models that can be seamlessly applied in both simulation and real world. Traditional approaches like domain randomization have demonstrated success in zero-short setting, by creating representations that are resilient and adaptable through the augmentation of diversity within simulations. However, they suffer from the need for extensive training across a range of parameter variances, and dependency on heuristic approaches. In this work, we present a novel reinforcement learning architecture named Soft Attention-Augmented Actor-Critic (Soft3AC) for sim-to-real robotic tasks without the need for heuristic domain randomization. Our approach achieves the learning of semantically task-relevant feature representations that exhibit resilience against appearance gaps. This is realized by employing an architectural design that separates current perceptions from historical perceptions in memory, fostering abstract spatial-temporal understanding. Simultaneously, the introduction of an attention mechanism enables a more contextual processing. We validated our method through conducting a valve rotation task with a robotic hand, under both sim-to-sim and sim-to-real conditions. The results indicate that our model adeptly bridges the appearance gap observed in sim-to-sim and sim-to-real transfers. Our method demonstrated its ability to be deployed directly into the real world in a domain randomization free zero-shot manner.**

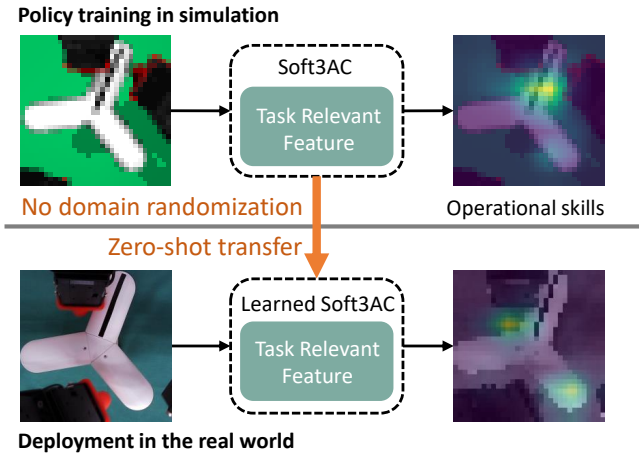


Fig. 1: Overview of the proposed method. Soft3AC learns task relevant feature during simulation, enabling a zero-shot sim-to-real transfer without domain randomization. We evaluated our method through a valve rotation task. The diagrams on the right indicate the operational skills learned as features during simulation training, which were successfully transferred to the real world.

I. INTRODUCTION

Reinforcement learning (RL) has seen significant advancements with the growth in its application to robotics. However, RL heavily relies on massive volumes of training data [1]. Real robot training is time-consuming and poses hardware damage risks from failed trials. One promising solution to these issues is sim-to-real transfer [2], which uses physics simulators to train and deploy on the physical hardware. However, these simulations, no matter how intricate or detailed, the simulation may not model some of the physical phenomena present in real world, is termed as sim-to-real gap [3].

To bridge this sim-to-real gap, it is critical to learn robust representation models that demonstrate resilience to alterations such as appearance gaps in lighting conditions, texture and perspective differences. In scenarios where real-world data are inaccessible, a common approach is the zero-shot method employing domain randomization [4], [5], [6], [7], [8], [9], [10] which generates a robust representation by augmenting the variety of conditions within simulations. However, it comes with its issues, such as needing extensive training over a range of parameter variances and dependency on heuristic approaches. Given these issues, our study seeks to create a robust representation model for robot tasks that is zero-shot sim-to-real transferable beyond the reality gap in appearance, without the need for extensive heuristically augmented data.

Learning representation models often involves deep learning, leveraging Neural Networks (NN) to distill features from data. Recently, these models have been benefiting notably from attention methods[11]. These attention methods are employed in transformers and large-scale language models and have exhibited success when dealing with substantial datasets for simulation-based tasks in navigation [12], [13], image recognition [14], [15] and language processing [16], [17] within simulated environments. However, their adaptability to the real world with reality gap has not been fully tested. Among these, the Attention-Augmented Memory (AAM) method [13] deserves notable mention. This method achieves acquiring high-level, task-relevant skills. These skills are less dependent on the specific attributes of the training

*This work was not supported by any organization
[1]J. Qu, S. Otsubo and S. Miwa are with the Advanced Technology R&D Center, Mitsubishi Electric Corporation, Hyogo, Japan kyoku.ka@dc.MitsubishiElectric.co.jp
[2]T. Yamanokuchi, and T. Matsubara are with the Graduate School of Information Science, Nara Institute of Science and Technology (NAIST), Nara, Japan yamanokuchi.tomoya.ys9@is.naist.jp

environment, offering a more robust representation model. The AAM method has proven its effectiveness in simulated navigation tasks, demonstrating considerable resilience to environmental variations.

In this study, we propose a novel RL architecture Soft3AC for sim-to-real robotic tasks without the need for domain randomization. Our method integrates a recurrent Actor-Critic RL framework with the AAM model. The architectural design distinctly separates current perceptions from past experiences stored in memory, incorporating an attention mechanism. The memory component archives a historical record of observations, promoting abstract spatio-temporal understanding rather than localized features. Furthermore, the attention mechanism discriminates crucial information by weighting memories and the current perception, allowing the agent to focus on shared contextual efficient features across different conditions. As a result, our RL architecture excels in learning high-level, semantically task relevant feature representations. These representations are less constrained by specific characteristics of the simulation, thus exhibiting enhanced generalizability to real-world scenarios. An overview of our method is shown in Fig.1.

We evaluated our method on a valve rotation task with a robotic hand in both sim-to-sim and sim-to-real scenarios. The results demonstrate that our model learns task-relevant operational skills as distinguishing features. In the sim-to-sim transfer scenario, our model displayed an impressive ability to bridge the appearance gap. Also, our model has demonstrated the potential to be deployed to real world beyond the appearance gap in a zero-shot manner without domain randomization.

Our contributions can be summarised as follows:

- We propose a RL architecture for zero-shot sim-to-real transfer of robotic tasks, without the need of domain randomization.
- We visually demonstrate that our method enables the learning of semantically task-relevant feature representations.
- We validate through experimental evaluations that our method improves performance in relation to the appearance gap in both sim-to-sim and sim-to-real settings.

II. Related Work

In this section, we provide an overview of previous work related to domain randomization for appearance gap in sim-to-real transfer.

Puang et al. [7] presented KOVIS, a keypoint-based visual servoing method for robotic manipulation tasks that achieves zero-shot sim-to-real transfer to real-world scenarios. The method learns keypoint representation from the camaera image with the training with data augmentation, domain randomization, and adversarial examples. However this keypoint-based method could misinterpret noise as keypoints and may struggle with texture-poor environments.

For a more robust feature extraction, James et al. [8] proposed Randomized-to-Canonical Adaptation Networks (RCANs), learns by converting randomized images into non-randomized, canonical counterparts. This technique has demonstrated significant efficiency in simulating real transfer on unseen objects and vastly decreased the dependence on real-world data.

Additionally, Yamanoguchi et al. [9] presented the Kalman Randomized-to-Canonical Model (KRC-model). Paralleling the concept of learning canonical features, this framework excels at deriving vital intrinsic features as well as their dynamic properties from randomized images. The KRC-model was subjected to the same valve rotation task as this study for evaluation. The experiment yielded that KRC-MPC can be effectively applied to diverse real-world domains and tasks, without any preliminary real-world input. Despite their merits, these models face challenges in parameter tuning for optimization specific to certain applications.

Seeking better tuning, Du et al. [10] proposed an auto-tuned sim-to-real transfer method that automatically tunes simulator system parameters to match the real world using only raw RGB images of the real world. This approach reframes the auto-tuning of parameters as a search problem. While demonstrating significant improvement over naive domain randomization, the process is time-consuming and computationally demanding.

Even though domain randomization has proven its effectiveness in addressing the appearance gap, including keypoint-based and canonical features learning, and auto-tuned adaptation, the problem of extensive training and tuning parameters remains unresolved. In our study, rather than immediately resorting to augmentation via domain randomization, we aim first to develop a robust RL algorithm in simulation that optimizes sim-to-real transfer. Our strategy addresses the standing issues with domain randomization, offering potential solutions to the inherent difficulties within current methods. Importantly, our approach does not discard the concept of domain randomization. Instead, it suggests its complementary use alongside our proposed RL algorithm. When combined, we anticipate that these approaches will deliver further enhanced performance.

III. Method

Our goal is to deploy the proposed Soft3AC model for sim-to-real robotic tasks without the need for heuristic domain randomization. This section will detail our architecture design, explain the function of the AAM module, and describe the learning mechanism we've implemented to achieve the formulation of a robust representation model.

A. Soft3AC Architecture

The architecture of our Soft3AC integrates the AAM model with the sturdy framework of Soft-Actor Critic (SAC) [18], as shown in Fig. 2. In a comparative comparison of three fundamental RL algorithms - DQN [19], PPO [20], and SAC by Mock et al. [21], it was observed that while all three algorithms demonstrated similar results, SAC uperior owing to its robustness and versatility. Consequently, we choose SAC to be the bedrock of our method. Under the basic SAC structure, AAM situates itself between the input observation

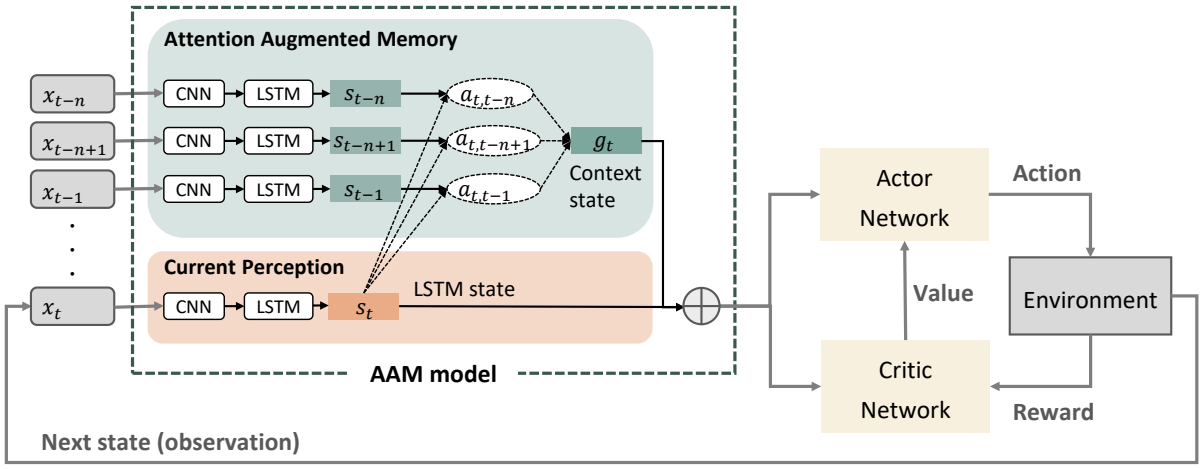


Fig. 2: Architecture of our proposed Soft3AC model, incorporating the AAM model within an Actor Critic framework. For each observation, a corresponding action is determined based on the state provided by the AAM through the Actor Critic network.

and the Actor/Critic network, replacing conventional CNN layers for extracting image features. Here, the output generated by AAM is a state which fuses the memory's context state with the current LSTM states. This resultant state is then fed into the Actor/Critic network, serving as the basis for policy training. Owing to SAC's inherent design as an off-policy actor-critic deep RL algorithm, Soft3AC adeptly strikes a balance between exploration and exploitation, leveraging the maximum entropy framework. This dual-component architecture employs an actor policy for action determination and a critic for anticipated return computation. In essence, the integrated SAC component introduces a temperature parameter which modulates the degree of randomness in action selection. With the prime focus on actions yielding higher expected returns, the algorithm also acknowledges actions generating lower returns but encapsulating greater uncertainty.

B. Attention-Augmented Memory

The AAM model, consists of two parts: attention-augmented memory and current perception. The attention-augmented memory module stores a fixed-size series of LSTM states of past observations. Using an attention layer, these past states and the current state are converted into a context state. The current perception module stores the current LSTM state. At every time step, the current perception module is updated and the attention-augmented memory module updates its context state correspondingly by sliding memory window from time step t-n to t-1.

AAM is a module that summarizes the context information of past states. The summarization is realized by storing historical LSTM states and implementing a selection mechanism through attention. Specifically, at each time step t, the CNN generates a representation mapr_t for the current observation x_t. Using the representation map r_t and the previous LSTM state s_{t-1}, LSTM then outputs a current state s_t. The trajectory memory maintains a historical LSTM state sequence s_{t-n},...,s_t, and transforms them into a context state by applying the content-based attention proposed in [22]. Context state g is defined as a weighted sum:

$$g = \sum_t \alpha_t s_t \tag{1}$$

$$e_t = w^T \tanh(W_\alpha s_t + b_\alpha) \tag{2}$$

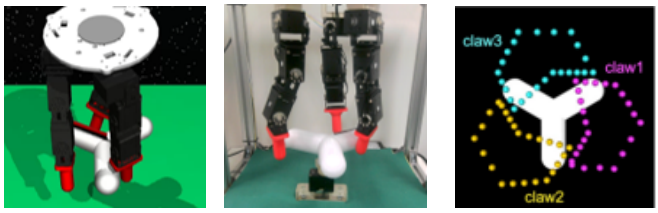
$$\alpha_t = \frac{\exp(e_t)}{\sum_i \exp(e_i)} \tag{3}$$

Here, W_a, w, $b_\alpha$are the model parameters.

C. Feature Learning Mechanism

Our primary objective is to learn generalized feature representation through the proposed Soft3AC architecture. In this architecture, the memory component permits the storage of the history of past observations. It learns to connect various states with specific outcomes by referring back to these experiences. This helps the system to develop an abstract understanding that moves beyond immediate localized image features and holistic representations reflecting broader patterns over time.

Adding to this, the attention mechanism weighs the importance of different parts of the memories. This allows the agent to focus more on relevant past experiences and less on irrelevant ones. Thus, it effectively learns to discriminate crucial contextual information for successful performance of the task. Notably, since the decision-making is learned by both the current perception and attention-weighted memories, the attention mechanism targets essential experiences and uses these as references for the model to adjust its interpretation of the current situation. This allows a more spatial-temporal contextual understanding of the task.



(a) (b) (c)

Fig. 3: Experiment setup. (a) ROBEL D'Claw simulation environment. (b) Real robot environment. (c) Action space.

By combining these elements together, the mechanism enables the model to focus on shared contextual efficient features' across varying conditions. Consequently, our model learns beyond low level, local image features, towards high-level task-relevant strategies and skills to solve the task that are less tied to the specific characteristics of the simulation, and can generalize better to real world situations.

IV. EXPERIMENTS

In this section, we describe the experimental setup, training procedures and the evaluation metrics.

A. Experimental Setup

- Valve Rotation Task: The goal of the Valve Rotation Task is to maintain a constant velocity (0.5rad/sec) while rotating the valve using three robotic fingers. We utilized both the ROBEL D'Claw simulation environment [23] and actual ROBEL D'Claw robot for this task. The simulation environment is built in the physics simulator MuJoCo [24] as shown in Fig. 3a. This environment consists of a 9- degrees-of-freedom (DoF) robot hand where each finger has 3-DoF and a 1-DoF valve. The real D'Claw robot, a hardware platform with identical settings as the simulation, is also depicted in Fig. 3b. In this task, a distinct black marker is present on the valve to differentiate its three sections. This is established as the default setting for the rotation functionality in this particular task.
- State Space: For the robot, each of its three fingers has three controllable joints, totaling nine dimensions across all fingers. For the valve, our focus was on enabling the model to identify the valve's position from image feature, hence, we do not utilize the joint angle of the valve. Therefore, the dimension of the state space stands at dz = 9.
- Action Space: We defined the action as the respective position of each fingertip and restricted them to reside within a one-dimensional (1D) manifold. In Fig. 3c, each colored dotted line corresponds to the manifold with each point on the lines representing a position. Due to the undesirability of discontinuity between 0 and 1 for model learning, the target positions of the controllable joints were used as actions during model learning, assigning du = 6. Actions were also normalized to the range [0, 1].
- Observation Space: A combination of a 32 x 32 RGB image and the encoder values of all finger joints (totaling 9D) were used as observations. We then set the dimension of the image feature at da = 8. This configuration resulted in a total observation dimension of dy = 17 for the model, consisting of both the image feature and encoder values of all finger joints.
- Reward function: The reward function serves as an indicator of how effectively the robot hand performs the valve rotation task, primarily focusing on minimizing the error distance at each step. It is defined as

$$r_t = -5\left|\Delta\theta_{\text{t, obj}}\right| + 10I\left(\left|\Delta\theta_{\text{t, obj}}\right| < 0.25\right) + 50I\left(\left|\Delta\theta_{\text{t, obj}}\right| < 0.1\right) \tag{4}$$

where $\Delta\theta_{\text{t,obj}}$ is the error between the goal and the current object at time t.

B. Training

The policy is trained in simulation using the simulated setup. For one episode, the task execution steps was set to T = 80. To manage the sequence of historical observation, we implement temporal attention with a size of 20. Experience replay is also used to enhance the efficiency of our training process. We store the agent's experiences in a replay buffer with a substantial size of 1e6. At each timestep of an episode, 256 mini-batches are randomly sampled from this replay buffer. These batches are then utilised to train both the policy network and two Q-function networks. The policy is represented by a neural network with hidden layer of size 256.

As a comparison baseline, we have also implemented the SAC without the AAM module. This comparison allows us to measure the efficacy of including the AAM module in our proposed model. We maintained identical settings during the training of this baseline policy to ensure a fair comparison. We trained a total of 10 models each for Soft3AC and SAC in the simulation environment.

C. Evaluation Metrics

Our experiments evaluate task performance via the tracking error and success rate metrics. We also visualize the saliency to evaluate what representation dose the model learn.

- Tracking Error and Success Rate. Basically we evaluate the tracking performance using tracking error (TE) defined by the following equation:

$$TE_t = \left|\theta_t^{\text{valve*}} - \theta_t^{\text{valve}}\right| \tag{5}$$

where $\theta^{\text{valve*}}$ and θ^{valve} refer to the target valve position and the actual valve position at time step t, respectively. In practical, in sim-to-real test, we evaluate a task if there is considerable failure through considerable failure rate (CFR)

$$CFR = \frac{\sum_{t=1}^{T} I(|\text{TE}(t)| > 1, rad)}{T} \tag{6}$$

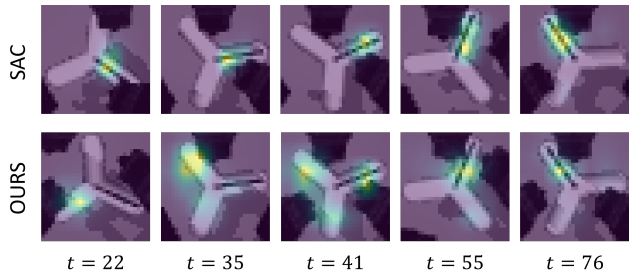


Fig. 4: Saliency in simulation. The representative images display saliency samples from selected steps in both SAC and our proposed method. Each image includes an overlaid heatmap on the original observation images, highlighting the areas of high focus.

where T is the total time step in one episode. Given multiple tests, we then calculate the overall Success Rate, which is simply the ratio of successful tests with CFR values less than the threshold (10%), over the total number of tests R performed:

$$SuccessRate = \frac{\sum_{r=1}^{R} I(|\text{CFR}(r)| < 0.1)}{R} \quad (7)$$

- Saliency Analysis. In order to show 'what' the robot is looking at in a current observation, we perform the saliency analysis presented in [25]. This perturbation saliency analysis introduces a local Gaussian blur at every single pixel in the image and measure the change in the policy. The change at each pixel forms the saliency map which can indicate how much the agent relies on every spatial area in the image to make decisions.

V. EXPERIMENTAL RESULTS AND DISCUSSION

The primary objective of our approach is to learn high-level task-relevant features from simulation that promote superior real world generalization without domain randomization. In this section, we present the learned representation features in simulation and adaptation results on sim-to-sim and sim-to-real transfer, comparing the performance of our method to the presented baseline.

A. *Observing Features in Simulation.*

We have utilized a saliency visualization to understand the representations learned from the simulation. This visualization highlights the significant areas within the current observation that are important for decision-making, as depicted in Fig.4.

As can be observed, the baseline SAC method consistently maintains its focus on the print marker on the valve, demonstrating no variations over time. It appears that SAC uses the immediate edge feature of the print marker as a means to predict and track rotation angles based on image observations. This implies that SAC is primarily learning the low-level object-relevant image features directly related to the assigned task.

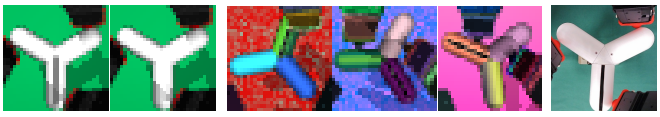
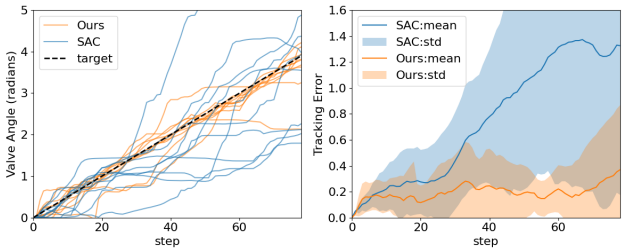


Fig. 5: Observation image samples in sim-to-sim and sim-to-real test. (a) Original setting. (b) No marker setting. (c) Random texture setting. (c) Observation of real robot.

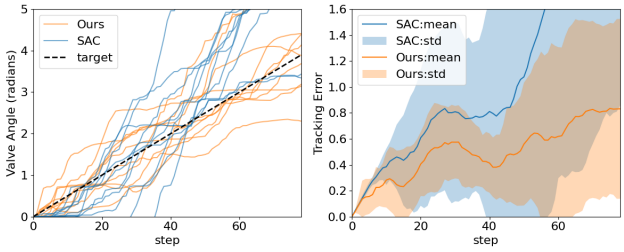
TABLE I: Evaluation results of sim-to-sim test. Average (Avg) and standard deviation (Std) of the tracking error over 10 random seeds model.

	No maker		Random Texture	
	Avg	Std	Avg	Std
SAC	0.75	0.6	1.2	1.0
Ours	**0.20**	**0.23**	**0.51**	**0.4**

In contrast, our proposed method demonstrates an adaptive focus shift towards the interaction areas between the robot's finger and the valve. This implies important evidence that our proposed method primarily learns task-relevant features of the key operational points that are crucial for successful task completion as well as for tracking angles. From these observations, we infer that our proposed model successfully constructs a higher level of temporal abstraction beyond the low-level localized image features that lead to cultivating a broader, more semantically relevant set of task-related feature representations.

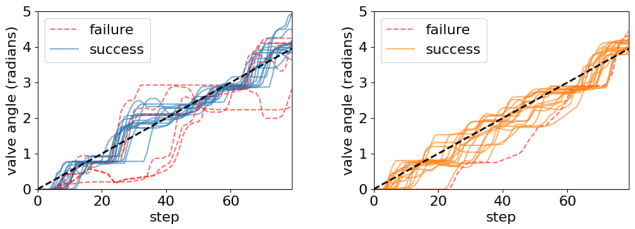


(a) No marker



(b) Random texture

Fig. 6: Tracking trajectories of sim-to-sim test without print marker. Top shows the valve angle at each step for all 10 random seeds models. Closer match to target indicates better task performance. Bottom displays average and standard deviation of tracking error at each step.



(a) SAC (b) Ours

Fig. 7: Tracking trajectories of sim-to-real test. It shows the valve angle at each step for all 10 random seeds models. Dotted red line shows instances with considerable failure.

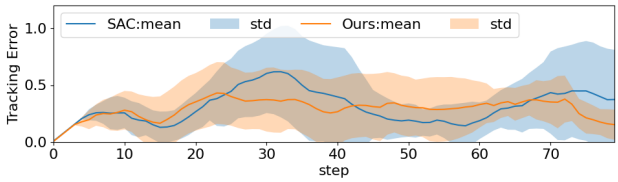


Fig. 8: Tracking error of sim-to-real test. It displays average and standard deviation of tracking error at each step.

B. Sim-to-sim Transfer Performance.

Before testing real world generalization, we tested generalization performance in sim-to-sim situations. We evaluated models trained on a specific simulation to different simulated conditions with appearance gaps. We added two types of appearance gaps. The observations gleaned from each setting are displayed in Fig.5.

The first appearance gap, referred to as the 'no marker setting', shown in Fig.5b, was devised by erasing the print marker of the valve, signifying a key change in the target object. The second appearance gap, termed as the 'random texture setting', shown in Fig.5c, was devised by randomizing the color and texture of both the valve objects and background. Fig.6a displays the adaptation results under the no marker setting, indicating that our method accurately tracked the target angle, even in the absence of the print marker. However, SAC's tracking performance diverged when the marker was removed.

The random texture setting introduced a significant appearance change. The outcomes presented in Fig.6b show that even with performance drops, our approach consistently tracked the target angle and surpassed SAC in reducing tracking errors as referenced in Table. I. In conclusion, our method proved to be resilient to the elimination of certain identities of the object itself since it learned task-relevant features without depending on local object-relevant visual features. Moreover, our technique displayed robustness against extensive appearance changes due to its adaptive focus on task-relevant regions, resulting in minimal environmental influence.

C. Sim-to-real Transfer Performance.

To test real world generalization, We evaluated the models trained in simulation on a real robot. The real world observation presents an appearance gap between simulation and real world, including lighting conditions, texture differences, and color deviations, as illustrated in Fig.5d. According to a comprehensive evaluation presented in Table.II and Fig.8, our method shows a reduction in tracking error and significantly fewer instances of considerable failure. From the tracking trajectories in Fig.7, both our model and SAC suffer from initial difficulties. However, their subsequent trajectories differ substantially.

Though initially our method displayed data points with large errors, it quickly recovered and stabilized during later phases. Our method demonstrates robust adaptability in the medium phase of operation, resulting in fewer failures over time. In contrast, although SAC also has some initial failures, it tends to encounter issues midway through the process in most cases. While initial failures are common and typically short-lived, the two methods diverge in terms of their ability to handle the gap in midway process.

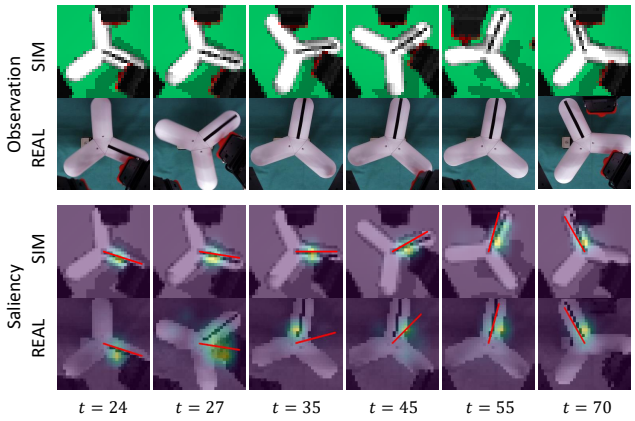
To understand how our method achieves this robust sim-to-real transfer, we examined the differential midway processes of the two methods. We compared the saliency areas of the simulated and real robots for the same models. As shown in Fig.9, for the baseline SAC, the real robot exactly mimics the policy seen in simulation in terms of finger movements throughout the task execution. However, due to the appearance gap, the ability to maintain focus on a specific region becomes challenging, leading to deviations in the robot's actions. This forces the agent to make several corrective attempts to regain sight of the targeted feature area (Fig.9a) or ends in fail if recovery is not possible (Fig.9b).

The saliency comparison of our proposed method is presented in Figure 10. In every test conducted, we notice that the real robot employs an adaptive policy to find the most appropriate solution. The focus area of the robot changes as per finger movements but consistently displays the interaction between the robot's finger and the valve. Whenever deviations occur, the robot is able to correct them in few steps (Fig.10a).

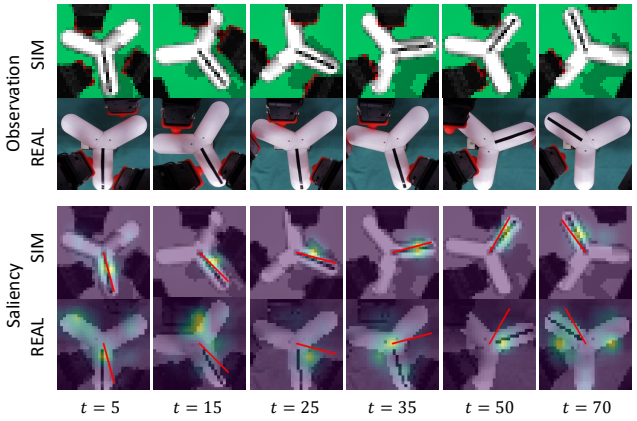
This distinction suggests that in SAC, the object's local visual feature, learned in simulation, exhibits instability during the sim-to-real transfer process. This leads to struggles in maintaining its adaptation over time, resulting in more midway failures. On the other hand, our method maintains stability in the learned high-level operational skill feature

TABLE II: Evaluation results of sim-to-real test. Average (Avg) and standard deviation (Std) of the tracking error and success rate over 10 random seeds model.

	Tracking Error		Success Rate
	Avg	Std	
SAC	0.31	0.24	78.9%
SAC+AAM	**0.29**	**0.21**	**94.1%**

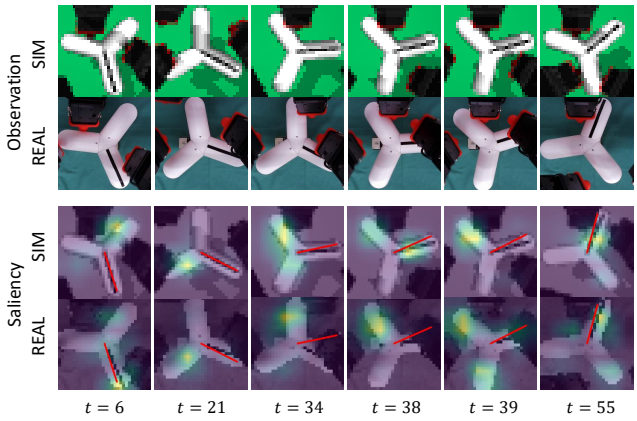


(a) test 1

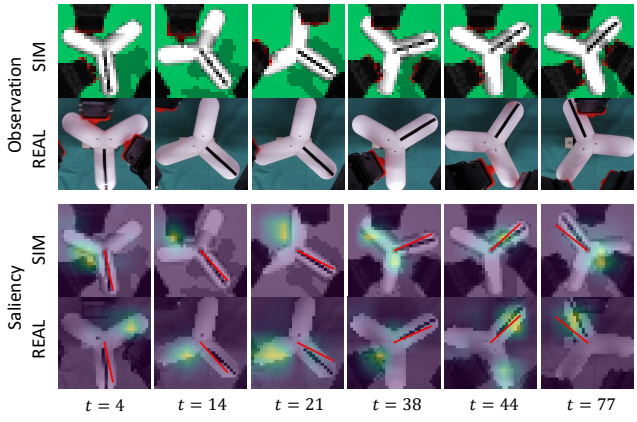


(b) test 2

Fig. 9: Observation and saliency in sim-to-real test: SAC. The red line indicates the target position. (a) Test 1 represents a comparison of simulated and real results for the same model. In the real test, the agent takes several corrective measures to retain focus on the specific feature (step 27 to 55). (b) Test 2 shows another set of outcomes where the real test ends in failure without being able to recover.



(a) test 1



(b) test 2

Fig. 10: Observation and saliency in sim-to-real test: Soft3AC. The red line indicates the target position. (a) Test 1 represents a comparison of simulated and real results for the same model. In the real test, the agent takes minimal corrective steps towards recovery (step 34 to 39). (b) Test 2 shows another set of results where the agent successfully executes the task.

representation during the sim-to-real transfer, demonstrating superior adaptation capability.

VI. CONCLUSIONS

In summary, this paper has presented a novel method for learning robust representation models pertinent to both sim-to-sim and sim-to-real robotic tasks using Attention-Augmented Memory. This approach addresses significant challenges of the sim-to-real gap in the field of robotics and offers an efficient solution without the need for heuristic data augmentation.

Our model separates current and past observations within memory and introduces an attention mechanism that enables the learning of task-relevant feature representations. We have demonstrated the adaptability and versatility of our method by its successful deployment on a valve rotation task performed by a robotic hand, under both sim-to-sim and sim-to-real conditions.

The results exhibit that our model effectively bridges the appearance gap observed in sim-to-sim and sim-to-real transfers. This proves the ability of our method to be deployed directly into the real world, showcasing a zero-shot transfer independent of domain randomization.

We consider this research to be a significant contribution to reinforcement learning in robotics, providing a new way of learning robust representation models that require no extensive data augmentation and can successfully perform in various application scenarios. In future works, we aim to explore further applications of Soft3AC in more complex dynamic environments and tasks, pushing forward the frontiers in the area of simulation-based reinforcement learning for real-world robotic applications.

REFERENCES

[1] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International journal of robotics research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[2] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 3803–3810.

[3] N. Jakobi, P. Husbands, and I. Harvey, "Noise and the reality gap: The use of simulation in evolutionary robotics," in *Advances in Artificial Life: Third European Conference on Artificial Life Granada, Spain, June 4–6, 1995 Proceedings 3*. Springer, 1995, pp. 704–720.

[4] F. Sadeghi and S. Levine, "Cad2rl: Real single-image flight without a single real image," *arXiv preprint arXiv:1611.04201*, 2016.

[5] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.

[6] J. Matas, S. James, and A. J. Davison, "Sim-to-real reinforcement learning for deformable object manipulation," in *Conference on Robot Learning*. PMLR, 2018, pp. 734–743.

[7] E. Y. Puang, K. P. Tee, and W. Jing, "Kovis: Keypoint-based visual servoing with zero-shot sim-to-real transfer for robotics manipulation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 7527–7533.

[8] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis, "Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 627–12 637.

[9] T. Yamanokuchi, Y. Kwon, Y. Tsurumine, E. Uchibe, J. Morimoto, and T. Matsubara, "Randomized-to-canonical model predictive control for real-world visual robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8964–8971, 2022.

[10] Y. Du, O. Watkins, T. Darrell, P. Abbeel, and D. Pathak, "Auto-tuned sim-to-real transfer," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 1290–1296.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[12] K. Fang, A. Toshev, L. Fei-Fei, and S. Savarese, "Scene memory transformer for embodied agents in long-horizon tasks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 538–547.

[13] J. Qu, S. Miwa, and Y. Domae, "Learning landmark-oriented subgoals for visual navigation using trajectory memory," in *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2022, pp. 708–714.

[14] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3286–3295.

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[16] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[18] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[19] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[20] L. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, *et al.*, "Model-based reinforcement learning for atari," *arXiv preprint arXiv:1903.00374*, 2019.

[21] J. W. Mock and S. S. Muknahallipatna, "A comparison of ppo, td3 and sac reinforcement algorithms for quadruped walking gait generation," *Journal of Intelligent Learning Systems and Applications*, vol. 15, no. 1, pp. 36–56, 2023.

[22] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

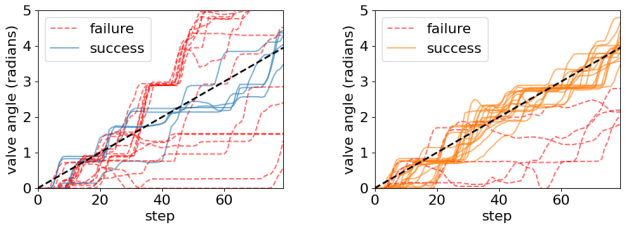
[23] M. Ahn, H. Zhu, K. Hartikainen, H. Ponte, A. Gupta, S. Levine, and V. Kumar, "Robel: Robotics benchmarks for learning with low-cost robots," in *Conference on robot learning*. PMLR, 2020, pp. 1300–1313.

[24] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.

[25] S. Greydanus, A. Koul, J. Dodge, and A. Fern, "Visualizing and understanding atari agents," in *International conference on machine learning*. PMLR, 2018, pp. 1792–1801.

APPENDIX

Here, we present additional results for the sim-to-real testing without the print marker. Similar to the test in sim-to-sim, we remove the print marker of the valve in real robot, representing a substantial appearance gap to the target object. Fig.11 and Fig.12 displays the sim-to-real transfer results under this no marker condition. The findings convey that our proposed model demonstrates remarkably fewer failures compared to SAC, even when the print marker is absent. This indicates superior adaptability and robustness of our method towards significant appearance in the task environment.



(a) SAC (b) Ours

Fig. 11: Tracking trajectories of sim-to-real test: No print marker. It shows the valve angle at each step for all 10 random seeds models. Dotted red line shows instances with considerable failure.

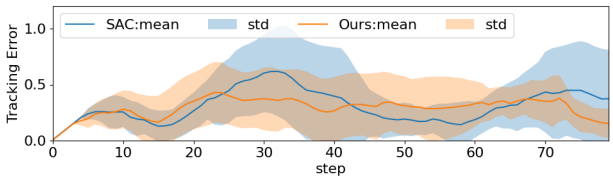


Fig. 12: Tracking error of sim-to-real test: No print marker. It displays average and standard deviation of tracking error at each step.