Continual Invariant Image Mapping from Randomized Simulations for Sim2real Transfer in Robotic Manipulation

Xiongyu Xie [1*], Josip Josifovski[1], Mohammadhossein Malmir[1] and Alois Knoll[1]

***Abstract*— Currently, deep reinforcement learning algorithms require large amounts of training data to learn a specific task, which makes them infeasible to train directly on real robotic systems. To overcome this obstacle, one usually relies on training in simulation and randomizes aspects of the simulation to compensate for the mismatch between the simulator and the real system. However, it is not always clear which aspect of the simulation requires randomization and usually enabling an additional randomization parameter or simulation modifications require model retraining from scratch. To address this problem, in this paper we explore how continual state representation learning can be combined with parameter randomization for vision-based reinforcement learning of robotic tasks, to minimize the need for complete model retraining. To this end, we use variational autoencoder (VAE) to continually learn to reconstruct invariant image representation from sequentially randomized/augmented simulation images. Independently, a reinforcement learning model is trained on the invariant image representation to solve a robotic manipulation task. Then, the VAE is used to translate randomized/augmented simulation images or real-world images to the invariant representation images on which the RL agent can operate. Initial results show that the VAE can continually learn reconstruction to invariant images and it can also be used to bridge the sim2real gap by reconstructing correctly real camera images.**

I. Introduction

Reinforcement learning in combination with deep learning has the potential to provide a general framework for learning-based robot control from raw input data, like camera images. However, drawbacks related to sample inefficiency of the current algorithms or safety implications when training trial-error approaches on real robotic systems lead to simulation-based training. While the shift to simulation-based training can circumvent the sample inefficiency or unsafe training problems, due to the mismatch between the simulated and the real system the applicability of the simulation-trained models remains challenging for real-world applications and widespread use in robot control. Methods like domain randomization [1] [2], [3] are often used to widen the training distribution with the expectation that the true parameters of the real system would be covered in the training phase, yet, there is no guarantee for success, and the widening of the randomization ranges can lead to sub-optimal performance [4].

On the other hand, continual learning methods are designed to work under the assumption that the underlying distribution might shift during training, that previous training data might become unavailable, and that the model should adapt to the new distribution during training without completely forgetting the previous data distributions. In this paper, we combine domain randomization with continual learning of invariant image representation for a specific robotic task, in order to investigate whether they can be used effectively for sim2real transfer and mitigate problems like changes in the underlying distribution or subsequent RL model retraining from scratch. To this end, we propose an approach that uses a variational autoencoder (VAE) to continually learn invariant image mapping from a randomized image observation in a given robotics simulator, where the visual input is augmented by different domain randomization techniques. Separately, a reinforcement learning agent is trained to solve a robotic task on the invariant (non-randomized) image input, in order to avoid the need for RL model retraining under changing environment and the risk of sub-optimal performance when the RL model is directly trained on randomized simulations with inappropriate randomization ranges. We evaluate the proposed architecture both in simulation environments with different simulation settings and in the real lab environment in terms of sim2real transfer. The preliminary results show that the combined state representation learning and reinforcement learning components can alleviate the need for RL model retraining and can help in smooth sim2real transfer.

II. Related Work

With the early success of deep reinforcement learning algorithms in video games [5] or simple 2D simulation environments like the ones provided in OpenAI Gym [6] it became more relevant to use robotic simulations and transfer RL agents from simulation to control real robotic systems. Since then, more complex robotic simulators and benchmarks [7], [8], [9] were introduced, however transferring models to real systems remains challenging. One direction to take is to make the simulated environment similar to the real environment to the greatest extent possible, e.g, by using high-quality rendering and the depth information of the image like in [10] or performing system identification [11] [12]. But there is a consensus that further improvements in simulation accuracy only cannot improve the sim2real transfer quality [13]. An alternative approach is to randomize parameters that cannot be precisely measured, simulated, or might vary w.r.t. the real system, like in [1] [2]. Building on domain randomization, James et al. [14] introduced Randomized-to-Canonical Adaption Networks to leverage neural networks to learn canonical representations of the actual environment from randomized environments. However,

* Corresponding Author: Xiongyu Xie xiongyu.xie@tum.de
[1] School of Computation, Information and Technology, Technical University of Munich, Germany.

naive randomization with inappropriate randomization ranges might hinder the learnability of a task [4]. To circumvent this problem, in [15] the authors use Automatic Domain Randomization, where the randomization increases gradually during training, making the training distribution wider and therefore the task more difficult only after the agent is performing well on the less-randomized setting. As not all randomization parameters are equally important, concepts like Active Domain Randomization [16] can reduce the cost of searching the most important and effective parameters that should be randomized.

Nevertheless, previous approaches assume that the simulator can already randomize all important parameters and that there is a strict separation between training and inference. On the other hand, continual learning, which can be applied in the context of robotics [17] relaxes these assumptions and offers a general framework to address changes in the data distributions, e.g. when the simulation or the task domain changes [18]. In this direction, defining the simulated and the real environment as two different tasks for an agent to learn, [19] uses continual learning for sim2real transfer and for effective reduction of the amount of training on the real system necessary during domain adaptation. Combining continual learning and state representation learning, [20] showed that VAE combined with generative replay achieves promising results when an agent continually learns navigation tasks in two different environments. Following that, the authors in [21] introduced a method to automatically detect the environment change, then allows VAE to self-trigger generative replay, which is more similar to the human way of learning. A deep generative replay framework is presented in [22] that is able to easily sample training data on previous tasks and interleave these data for new tasks.

III. Methodology

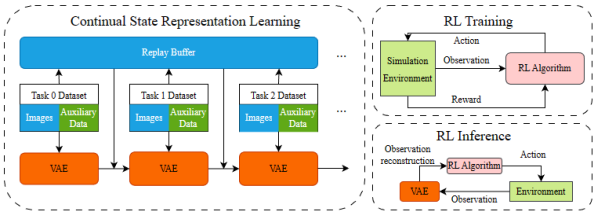

Fig. 1. The proposed model structure. On the left side of the figure VAE trained using continual learning with different randomized image datasets is shown. On the top right side the reinforcement learning setup for training is shown, where the agent is trained on non-randomized simulation. On the bottom right the inference phase is shown, where the observation for the RL agent is processed by the VAE to reconstruct non-randomized image from a real-world image on which the RL agent can operate.

The problem we want to solve can be described as the following: Given a real-world camera image as an input, a state representation learning model pretrained in simulation should reconstruct an invariant image representation, which is suitable for an RL agent trained in simulation to solve a robotic manipulator task. In the same time, the need for retraining any part of the model from scratch should be minimized in cases when simulation modifications, finetuning or domain adaptation with real world images is needed. To achieve this, we propose combining domain randomization and continual state representation learning components with a reinforcement learning agent trained on non-randomized simulations as described below and presented in Figure 1.

A. Domain randomization

We define datasets of non-randomized simulation images or simulation images with different types of randomization applied to them as different tasks. A state representation model should learn reconstructions from these datasets to a non-randomized simulation image continually, as shown in Figure 2. The Task 0 dataset consists of the original simulaton images obtained directly from the top-view camera without randomization. The Task 1 dataset images are based on the original images with the addition of the randomization w.r.t. the camera pose, background colors and scene lighting. The Task 2 dataset is generated by post-processing the non-randomized images and applying different types of noise (whitenoise, saltNoise, posterize and sharpness) using the Imagaug library [23]. For each task, the target image to be reconstructed is the original (not-randomized) image.

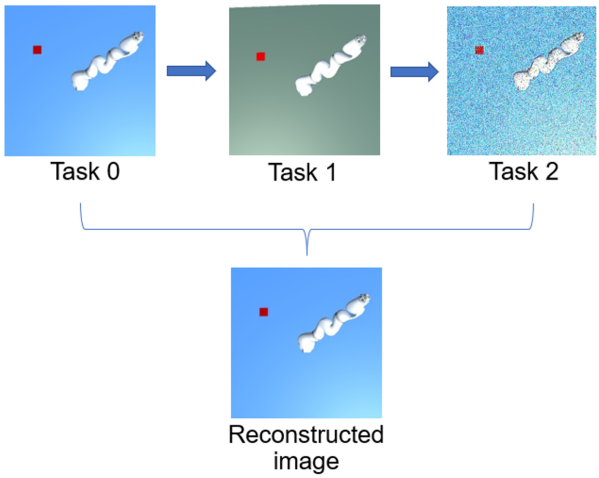

Fig. 2. Sample images from the different datasets used for training the State Representation Learning Model.

B. Continual State Representation Learning

Variational autoencoders (see Fig. 3) are used to learn the state representation from the simulation image datasets. Additional fully connected layers are used to reconstruct auxiliary input (e.g., target coordinates for the target reaching task) from the embedding vectors, which helps to reconstruct task-specific image details better. The loss function of the VAE is defined as

$$Loss = \alpha loss_{reconstruction} + \beta loss_{auxiliary} + \gamma Loss_{KL}, \quad (1)$$

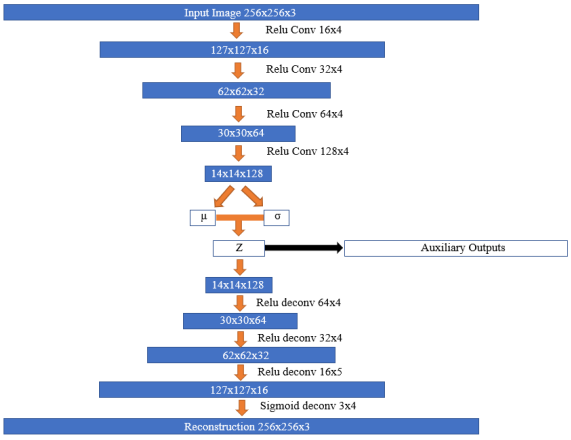

Fig. 3. The structure of the variational auto-encoder.

where α, β and γ are the hyperparameters to be tuned. KL annealing is applied to help VAEs to learn to reconstruct the images in the early stages of training [24].

Two replay-based continual learning approaches are implemented to learn the state representation of the images continually:

1) Experience replay (ER) [25].
2) Dark experience replay (DER) [26].

In ER, sample data from past tasks is repeatedly presented to the state representation model when the model is trained on a new task. As en extension of ER, DER attempts to encourage the learning model to mimic its original responses for past samples by adding an extra loss term, thus promoting consistency between current and past models. For DER, in addition to sample data from past tasks, it requires also the previous task models to be kept in memory.

C. The Complete Model Structure

The complete model consists of one VAE and one RL model, as shown in the bottom right corner of Fig. 1. The VAE is trained from Task 0 to Task 2 to learn the non-randomized image representation from the images. The RL model is trained in simulation to perform robotic reaching tasks from simulated non-randomized images. At inference time, the input to the VAE is switched to real camera images, whose reconstructions are then passed to the RL agent as observations.

IV. Experiments

Firstly, the comparisons between different VAEs on the three task datasets are reported in terms of invariant image reconstruction quality. Then the performance of the VAEs on the RL agent is evaluated by combining the same RL model with different VAEs. Finally, the evaluation of the sim2real transfer is conducted.

A. VAE Reconstruction Evaluation

To compare the VAE-based continual state representation learning models, three baseline models are trained to demonstrate the performance, advantages, and disadvantages of continual learning models. These models are described as follows:

- **Vanilla VAE for a single task:** Ideally, training a new VAE model separately for each task dataset allows the most accurate reconstruction of the specific task. As one of the baseline models, vanilla VAEs for each task separately are trained, to potentially give the upper limit of the VAE model used for image data reconstruction.
- **Fine-tuning VAE:** As a naive continual learning baseline, the fine-tuning model uses weights of the trained model from previous task as initialization for the next task. Here, fine-tuning VAE means taking the VAE trained on task 0 and then continuing with the training on Task 1 only and subsequently on Task 2 only images.
- **Joint training VAE:** Joint training VAE means that the data from all tasks is used at the same time during the training. Note that for joint training, each dataset from different tasks is sub-sampled and then combined to make a joint training dataset.

Each task dataset contains 10,000 images in which only the first two joints of the robot arm can move and the target is placed at a random position. All VAE models are trained for a total of 450 epochs. For the fine-tuning VAE, the VAE is trained firstly for 250 epochs on Task 0 to converge, then the model is fine-tuned on Task 1 and on Task 2 for 100 epochs each. For the continual learning (ER or DER) training, the same VAE trained on task 0 with 250 epochs is used as the starting model, and then trained for 100 epochs on Task 1 and 100 epochs on Task2 respectively to reach a total of 450 epochs. The joint training dataset also contains 10,000 images, with each of the three tasks accounting for one-third of the dataset.

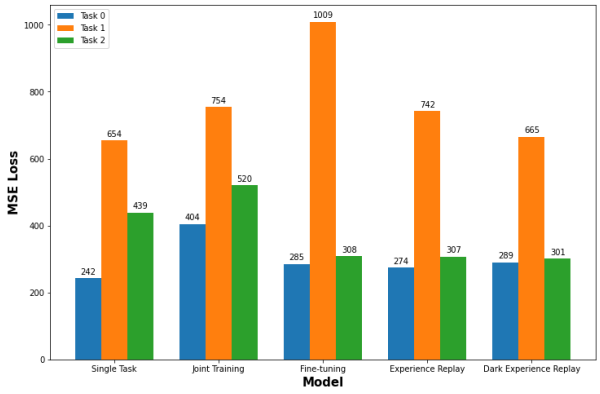

Fig. 4. Evaluation of the VAEs by the mean square error. Each model is evaluated on three test datasets for the three defined tasks. The single task means the model is trained on a single dataset only. The MSE loss is calculated by accumulating the loss on every pixel of the test dataset. Notice that, these three test datasets share a common ground truth dataset.

We evaluate every VAE model quantitatively on three test datasets, which correspond to the three tasks (see Fig. 4). As expected, the models trained on a single dataset have generally the best performance, except the model for Task 2. We hypothesize that this might be due to the task

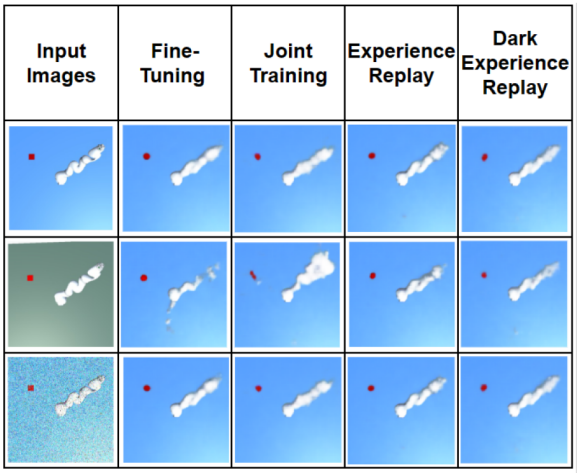

Fig. 5. Reconstruction of sample test images from the three tasks with the different VAEs trained on all tasks. Rows from top to bottom represent test images from Task 0 to Task 2 respectively.

specifications, namely it might be easier for the model to learn to reconstruct invariant image representations from noise-free images initially, compared to a training where noise-enabled (Task 2 images) are available from the start. Another interesting observation is that the finetuning model and continual learning models have similar performances on Task 0, but the fine-tuning model reconstructs the images much worse than the other two models on Task 1 due to "catastrophic forgetting", while all three models perform about the same on Task 2. On Task 2, DER has a better performance than any other model. Qualitative results of the reconstruction for sample test images from each task are presented in Figure 5.

B. Sim2real Transfer Evaluation

To evaluate whether the VAE reconstructions can be used as inputs to an RL agent trained on non-randomized images, we use a setup similar to [4] to train an RL agent for target reaching task, where the agent should learn to control the joint velocities of the robot's first 2 joints, using RGB images as observation and the negative distance of the end-effector to the target as reward. The RL agent is trained on non-randomized images only. In our experiments we use the Proximal Policy Optimization (PPO) [27] algorithm implementation from StableBaselines3 [28].

The preliminary results show that reconstruction images of the VAE trained on task 0 (non-randomized images) only show that this model cannot reconstruct a realistic image and it tends to output similar robot arm reconstruction regardless of the input image, although it can reconstruct the target location correctly (see Figure 6). As a consequence, the robot arm keeps making meaningless swings. On the other hand, the joint training VAE reconstructs the real images as the images in the simulator, but the robot fails to reach the target. VAEs trained sequentially on all three datasets using the continual learning (ER and DER) methods reconstruct the images better which helps the robot reach the target.

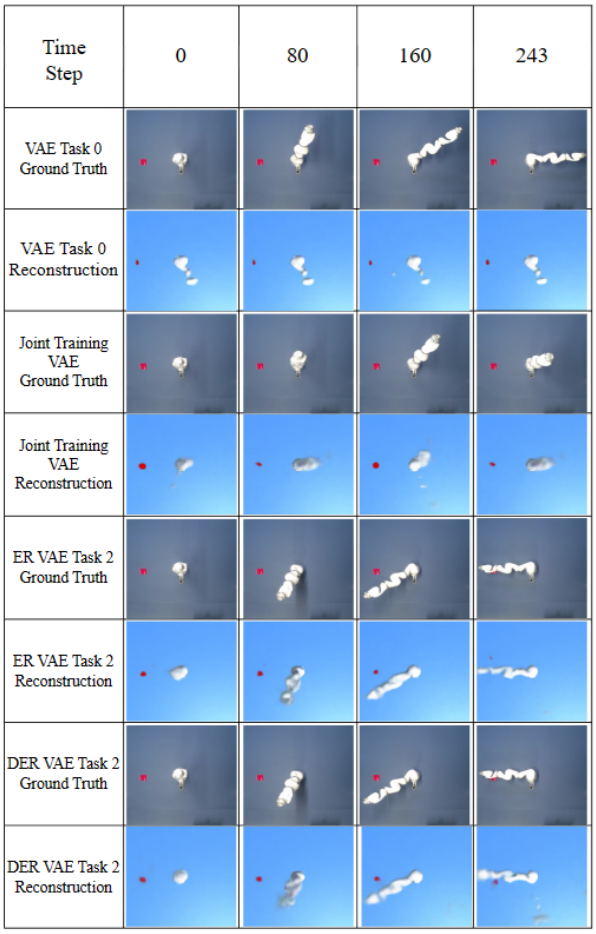

Fig. 6. Reconstruction of the real images in several time steps. VAE task 0 reconstructs meaningless images. The other three models can reconstruct the images similar to the real images, where the robot arm and the target are correctly rebuilt.

REFERENCES

[1] 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)*, 2017, pp. 23–30.

[2] J. Josifovski, M. Kerzel, C. Pregizer, L. Posniak, and S. Wermter, "Object detection and pose estimation based on convolutional neural networks trained with synthetic data," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 6269–6276.

[3] F. Muratore, F. Ramos, G. Turk, W. Yu, M. Gienger, and J. Peters, "Robot Learning From Randomized Simulations: A Review," *Frontiers in Robotics and AI*, vol. 9, p. 799893, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frobt.2022.799893/full

[4] J. Josifovski, M. Malmir, N. Klarmann, B. L. Žagar, N. Navarro-Guerrero, and A. Knoll, "Analysis of randomization effects on sim2real transfer in reinforcement learning for robotic manipulation tasks," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 10 193–10 200.

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with Deep Reinforcement Learning," in *NIPS: Deep Learning Workshop*, 2013.

[6] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," 2016.

[7] E. Falotico, L. Vannucci, A. Ambrosano, U. Albanese, S. Ulbrich, J. C. Vasquez Tieck, G. Hinkel, J. Kaiser, I. Peric, O. Denninger, N. Cauli, M. Kirtay, A. Roennau, G. Klinker, A. Von Arnim, L. Guyot, D. Peppicelli, P. Martínez-Cañada, E. Ros, P. Maier, S. Weber,

M. Huber, D. Plecher, F. Röhrbein, S. Deser, A. Roitberg, P. van der Smagt, R. Dillman, P. Levi, C. Laschi, A. C. Knoll, and M.-O. Gewaltig, "Connecting Artificial Brains to Robots in a Comprehensive Simulation Framework: The Neurorobotics Platform," *Frontiers in Neurorobotics*, vol. 11, no. 2, 2017.

[8] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning," in *Conference on Robot Learning (CoRL)*, vol. 100. Virtual Event: PMLR, 2020, pp. 1094–1100.

[9] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac Gym: High Performance GPU Based Physics Simulation for Robot Learning," in *Conference on Neural Information Processing Systems (NeurIPS)*, ser. Datasets and Benchmarks Track, Virtual Event, 2021.

[10] B. Planche, Z. Wu, K. Ma, S. Sun, S. Kluckner, O. Lehmann, T. Chen, A. Hutter, S. Zakharov, H. Kosch *et al.*, "Depthsynth: Real-time realistic synthetic data generation from cad models for 2.5 d recognition," in *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017, pp. 1–10.

[11] S. James and E. Johns, "3D Simulation for Robot Arm Control with Deep Q-Learning," in *NIPS Workshop: Deep Learning for Action and Interaction*, Barcelona, Spain, 2016.

[12] M. Kaspar, J. D. Muñoz Osorio, and J. Bock, "Sim2Real Transfer for Reinforcement Learning Without Dynamics Randomization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA, 2020, pp. 4383–4388.

[13] S. Höfer, K. Bekris, A. Handa, J. C. Gamboa, F. Golemo, M. Mozifian, C. Atkeson, D. Fox, K. Goldberg, J. Leonard, C. K. Liu, J. Peters, S. Song, P. Welinder, and M. White. Perspectives on Sim2Real Transfer for Robotics: A Summary of the R:SS 2020 Workshop. [Online]. Available: http://arxiv.org/abs/2012.03806

[14] 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. [Online]. Available: http://arxiv.org/abs/1812.07252

[15] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas *et al.*, "Solving rubik's cube with a robot hand," *arXiv preprint arXiv:1910.07113*, 2019.

[16] B. Mehta, M. Diaz, F. Golemo, C. J. Pal, and L. Paull. Active Domain Randomization. [Online]. Available: http://arxiv.org/abs/1904.04762

[17] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez, "Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges," *Information fusion*, vol. 58, pp. 52–68, 2020.

[18] J. Josifovski, M. Malmir, N. Klarmann, and A. Knoll, "Continual learning on incremental simulations for real-world robotic manipulation tasks," 2020.

[19] A. A. Rusu, M. Večerík, T. Rothörl, N. Heess, R. Pascanu, and R. Hadsell, "Sim-to-real robot learning from pixels with progressive nets," in *Conference on robot learning*. PMLR, 2017, pp. 262–270.

[20] H. Caselles-Dupré, M. Garcia-Ortiz, and D. Filliat. Continual State Representation Learning for Reinforcement Learning using Generative Replay. [Online]. Available: http://arxiv.org/abs/1810.03880

[21] H. Caselles-Dupre, M. Garcia-Ortiz, and D. Filliat, "S-TRIGGER: Continual State Representation Learning via Self-Triggered Generative Replay," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–7. [Online]. Available: https://ieeexplore.ieee.org/document/9533683/

[22] H. Shin, J. K. Lee, J. Kim, and J. Kim. Continual Learning with Deep Generative Replay. [Online]. Available: http://arxiv.org/abs/1705.08690

[23] I. Kostrikov, D. Yarats, and R. Fergus. Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels. [Online]. Available: http://arxiv.org/abs/2004.13649

[24] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. Generating Sentences from a Continuous Space. [Online]. Available: http://arxiv.org/abs/1511.06349

[25] L.-J. Lin, "Self-improving reactive agents based on reinforcement learning, planning and teaching," *Machine learning*, vol. 8, pp. 293–321, 1992.

[26] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and g.-i. family=CALDERARA, given=SIMONE, "Dark Experience for General Continual Learning: A Strong, Simple Baseline," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 15 920–15 930. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/b704ea2c39778f07c617f6b7ce480e9e-Abstract.html

[27] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[28] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html