Is Self-Supervision Helpful? Exploring Self-Supervision in Reinforcement Learning and Classification

Prabhat Agarwal, Abhishek Sinha, Kumar Ayush
{prabhat8, a7b23, kayush}@cs.stanford.edu

I. Abstract

Human labels are expensive to collect and hard to scale up. To this end, there has been an increasing research interest to investigate learning representations from unlabeled data. In particular, the image itself already contains structural information which can be utilized. Self-supervised learning approaches attempt to capture this. Self-supervised tasks such as rotation task loss as an auxiliary loss function have shown to improve classification performance across a range of classification datasets. In this work, we aim to explore the benefits of such self-supervision in two classes of tasks: (a) off-policy reinforcement learning, (b) classification (e.g. detection of forgery in manipulated facial images). Code can be found at this link.

II. Introduction

Our goal is to explore the effects of self-supervision in classification and off-policy reinforcement learning. To do this, in addition to supervised losses, we consider self-supervised losses based on labeled data $x \rightarrow (\hat{x}, \hat{y})$ that can be systematically derived from inputs x alone. We aim to consider self-supervised tasks like rotation task or sequence prediction task in our work. The rotation task uses the angle of the rotated image as the target label. Sequence Prediction is an auxiliary task where the sequence of the input frames, in reinforcement learning, is permuted and the network is tasked to identify the class of the permutation. A separate function h is used to predict these labels from the shared feature backbone f. Our final loss function will combine the two losses and thus the self-supervised losses act as a data-dependent regularizer for representation learning. We also explore the effects of self-supervised representation learning via Bidirectional GANs [1] in downstream classification tasks.

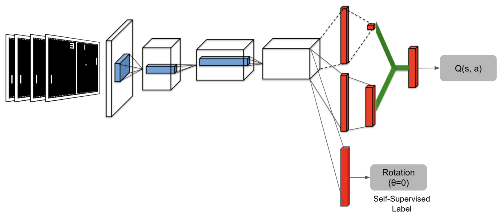


Fig. 1: Self-supervision in Pong with angle of rotation prediction.

We incorporate the aforementioned self-supervision tasks in (a) off-policy reinforcement learning, and (b) classification. An example of self-supervision in each class of task is shown in Figures 1, 2, and 3.

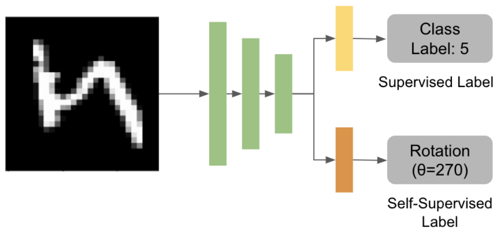


Fig. 2: Self-supervision in MNIST classification with angle of rotation prediction.

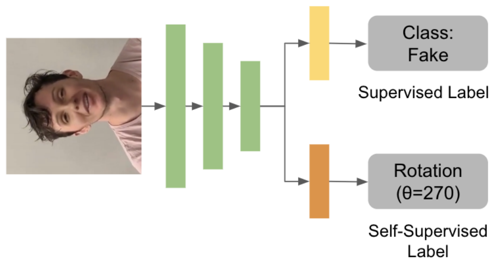


Fig. 3: Self-supervision in DeepFake classification with angle of rotation prediction.

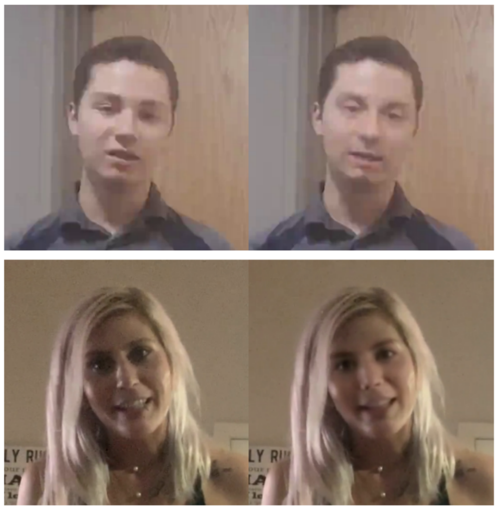


Fig. 4: Some examples from DeepFake Detection Challenge Dataset.

III. Related Work

There is a rich line of work on self-supervised learning. One class of methods remove part of the visual data (e.g., color information) and tasks the network with predicting what has been removed from the rest (e.g., greyscale images) in a discriminative manner [2], [3]. Su et al. [4] showed that self-supervision improves the transferability of representations for few-shot learning tasks on a range of image classification datasets. They introduced self-supervised tasks such as rotation task loss, jigsaw puzzle task loss as auxiliary loss functions. Recently, there has been a flurry of activity ([5], [6], [7], [8], [9], [10]) aiming to study the representations learned via self-supervision over ImageNet dataset without any labels. The most common evaluation protocol is the linear evaluation, where a linear layer is trained over the labeled Imagenet dataset on top of the frozen representations.

Contrary to these works, which have shown the usefulness of self-supervised representation learning over tasks that use data similar to the one used to learn the representation, we explore the robustness of the representations over out-of-domain datasets. To simulate this domain shift, we learn representations over ImageNet [11] dataset, but test the representations over the DeepFake dataset.

We further explore the usefulness of self-supervised learning in the domain of Reinforcement Learning.

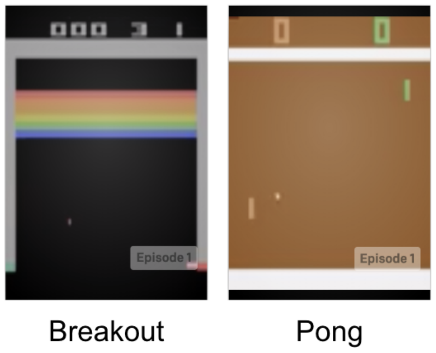


Fig. 5: We perform our experiments on two Atari Games: Pong and Breakout. We use the open AI gym simulator for the game.

IV. Modelling

Our framework augments standard supervised losses with those derived from self-supervised tasks to improve representation learning as seen in Figure 1 and 2.

A. *Classification*

In classification, we are given a training dataset $\{(x_i, y_i)\}_n^{i=1}$ consisting of pairs of images $x_i \in X$ and labels $y_i \in Y$. A feed-forward CNN $f(x)$ maps the input to an embedding space, which is then mapped to the label space using a classifier g. The overall prediction function from the input to the labels can be written as $g \circ f(x) : X \to Y$. Learning consists of estimating functions f and g that minimize a loss l on the predicted labels on the training data, along with suitable regularization R over the functions f and g, and can be written as:

$$L_s := \sum_i l(g \circ f(x_i), y_i) + R(f, g) \tag{1}$$

On tasks we assume that the test set consists of novel images from the same distribution as the training set. We use the standard cross-entropy (softmax) loss computed over posterior label distribution predicted by the model and target label.

For classification, we consider two kinds of self-supervision:

- In addition to supervised losses, we consider self-supervised losses L_{ss} based on labeled data $x \to (\hat{x}, \hat{y})$ that can be systematically derived from inputs x alone. Figure 2 shows an example of the rotation task that uses the angle of the rotated image as the target label. A separate function h is used to predict these labels from the shared feature backbone f and the self-supervised loss can be written as:

$$L_{ss} := \sum_i l(h \circ f(\hat{x_i}), \hat{y_i}) \tag{2}$$

We use rotation task loss as the self-supervised loss in our experiments. In this case, the input image x is rotated by an angle $\theta \in \{0°, 90°, 180°, 270°\}$ to obtain $\hat{x}$ and the target label $\hat{y}$ is the index of the angle.
Our final loss function combines the two losses:

$$L := L_s + L_{ss} \tag{3}$$

and thus the self-supervised losses act as a data-dependent regularizer for representation learning.

- We also leverage self-supervised representation learning using Bidirectional GAN [1] (trained with ImageNet [11]) and train a classifier over the learned features (See Figure 7). In contrast to the normal GAN framework, which consists of a Generator and a Descriminator, Bidirectional GAN also consists of an Encoder network. The role of the Encoder network is to map the high dimensional image representation to a low dimensional latent representation. The three networks are trained in an adversarial fashion, where the discriminator outputs a real/fake probability on the basis of a pair of image and latent vector as the input. The encoder of the BiGAN predicts these semantic latent representations which may serve as useful feature representations for downstream tasks where the learned semantic representations are useful. We demonstrate that the resulting learned feature representations are useful for downstream tasks like classification and is competitive with existing supervised learning approaches.

B. Reinforcement Learning

Reinforcement learning optimizes policies for expected cumulative reward. Reward is delayed and sparse for many tasks, making it a difficult and impoverished signal for end-to-end optimization. To augment reward, we consider self-supervised tasks to provide auxiliary losses. These losses offer instantaneous supervision for representation learning even in the absence of reward. While current results show that learning from reward alone is feasible, pure reinforcement learning methods are constrained by computational and data efficiency issues that can be remedied by auxiliary losses. We aim to explore whether self-supervised pre-training and joint optimization improve policy returns of end-to-end reinforcement learning as shown in Figure 6.

For reinforcement learning tasks, we use two kinds of self-supervision:

- *Image Based Self-Supervision (Rotation)*: Each input state in a sampled batch is rotated by an angle $\theta \in \{0°, 90°, 180°, 270°\}$ to obtain $\hat{x}$. In addition to predicting the $Q(s, a)$, the network is also tasked to predict the angle of rotation of the input state (See Figure 1).
- *Sequence Based Self-Supervision*: The input state to a Duelling Deep Q-learning Network consists of a concatenation of four consecutive image frames. We set up an auxiliary task where sequence of the four input frames, $x = (f_1, f_2, f_3, f_4)$, is permuted using a permutation α to obtain $\hat{x}$. With input $\hat{x}$, the network is tasked to identify the class of the permutation that when applied to a valid sequence produces the input sequence. We use the following permutations with the corresponding class label:
  - Class 0: $[0, 1, 2, 3], [3, 2, 1, 0]$
  - Class 1: $[0, 2, 1, 3], [3, 1, 2, 0]$
  - Class 2: $[0, 3, 2, 1], [1, 2, 3, 0]$
  - Class 3: $[1, 3, 0, 2], [2, 0, 3, 1]$

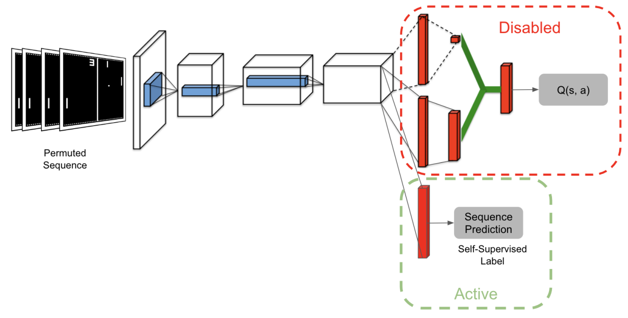
  Since $rev(x)$ and x are both valid input sequences, and $\alpha(rev(x)) = (rev(\alpha))(x)$, we group α and $rev(\alpha)$ in the same class for prediction.
  We alternate between $Q(s, a)$ prediction and sequence prediction during training (See Figure 6).

V. Experiments

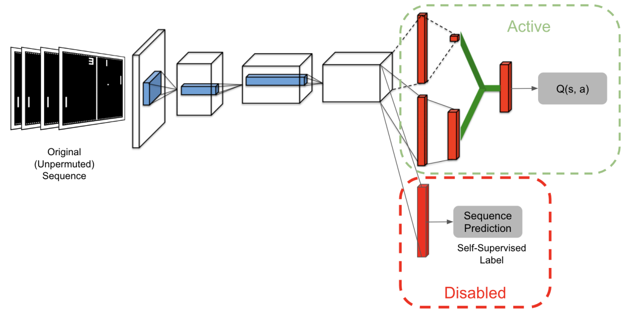
A. Dataset

1) Classification:

- **MNIST**: We performed our initial experiments on MNIST. MNIST dataset consists of



(a) The sequence of frames in the input state is permuted and the network is tasked to predict the correct sequence. Note that the $Q(s,a)$ prediction module is disabled.



(b) The sequence of frames in the input state is not changed during $Q(s,a)$ prediction. Note that the sequence prediction module is disabled.

Fig. 6: Self-supervision in RL using sequence prediction. We incorporate self-supervision in Dueling Deep Q-Learning algorithm. In this variant, we alternate between $Q(s,a)$ prediction and sequence prediction during training.

a train/test split of 60K/10K images. Each example is a 28×28 grayscale image associated with a lable from 10 classes.

- **DeepFake Detection Challenge Dataset**: The rapid progress in synthetic image generation and manipulation has now come to a point where it raises significant concerns for the implications towards society. Modern generative models are one example of these, capable of synthesizing hyperrealistic images and videos. Deepfakes—produced by deep generative models that can manipulate video and audio clips—are one of these. With such advances in computer vision, it is now possible to generate videos with extremely realistic synthetic faces, even in real-time. But this could be harmful to individuals and society and poses serious issues. Thus, there is a need for new methods to mitigate the potential for harm and abuse. Our work advocates research in this area and aims to leverage self-supervised learning for the same. Facebook introduced the Deepfakes Detection Challenge (DFDC) dataset (see Figure 4) consisting of 5K videos featuring two facial modification algorithms. We sample 368 real videos and 400 manipulated videos from the dataset. We created two train/test splits from the sampled dataset: (a) Random Split: Frames from a real or manipulated video may be in both train and test resulting in information leakage, (b) Hard Split: All frames of a video are either in train or test. We run face detection over the image frames to detect the bounding boxes corresponding to the face, and then only use the cropped face image to perform our experiments.

2) *Reinforcement Learning:* : We perform our experiments on two Atari (see Figure 5) games: Pong and Breakout. We use the open AI gym simulator for the game. The input states are four consecutive frames resized to 84×84 and converted to grayscale before being fed to the network. The action space in Pong is discrete corresponding to 6 different actions. The action space in Breakout is discrete corresponding to 4 different actions.

B. *Network Architecture*

1) *Classification:*

- **MNIST**: The shared backbone network consists of 2 CNN layers followed by a max-pooling layer of size 2×2 and a fully connected layer of size 128. The class prediction module is a fully connected layer of size 10 corresponding to the 10 classes. We use rotation task for self-supervision in this case and the rotation prediction module consists of a fully connected layer of size 4 corresponding to the 4 angles of rotation ($\theta \in \{0°, 90°, 180°, 270°\}$). See TABLE V for the architectural details.
- **DeepFake Detection Challenge Dataset**: We train binary detection classifiers over ResNet-50 [12] and AlexNet [13] backbones with various initializations, (a) ResNet-50 backbone with random initialization, (b) ResNet-50 backbone pre-trained with ImageNet [11], (c) AlexNet backbone pre-trained with ImageNet,

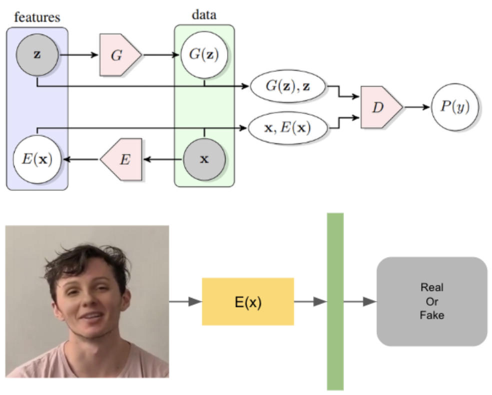


Fig. 7: Self-Supervised representation learning using Bidirectional GAN (pre-trained with Imagenet).

(d) AlexNet backbone pre-trained with ImageNet and rotation classification. We also train classifiers over features obtained from the encoder of an ImageNet pre-trained Bidirectional GAN (BigBiGAN [14]), (a) BigBiGAN with ResNet-50 [12] as the Encoder, (b) BigBiGAN with RevNet-50X4 [15] as the Encoder. For all the network architectures, we keep the backbone fixed, and only train a classifier on top of the features obtained from the backbone. This ensures that the representation learning performance is solely guided by the features extracted by the network, and thus other network hyperparameters do not affect the analysis. Through these experiments we want to separate the contribution of self-supervision in learning feature representations and highlight that the feature representations from the backbone contain useful information that can be generalized to a new out of distribution dataset and withstand significant domain shift.

2) Reinforcement Learning: We incorporate self-supervision in state-of-the-art Dueling Deep Q-Learning algorithm to estimate $Q(s,a)$ value for a $(state, action)$ pair. Our network consists of 3 CNN layers, followed by 3 fully connected layers in the shared backbone followed by task specific layers for $Q(s,a)$ prediction and auxiliary task prediction (See Figure 1 and 6).

C. Evaluation Metric

Classification: We use classification accuracy on the test set to assess the performance of the model.

Model	Accuracy
MNIST	99.19
MNIST with Self-Supervision	98.45

TABLE I: Overall classification accuracy of MNIST (on test data) when the network is trained with and without self-supervision.

Reinforcement Learning: We use Mean Score in 100 episodes for evaluating the performance of the trained RL agent.

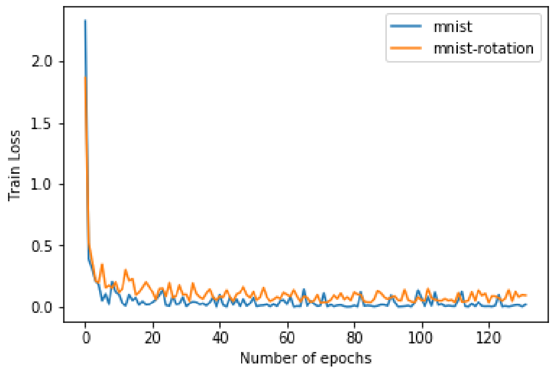
VI. Discussion and Analysis

A. Classification

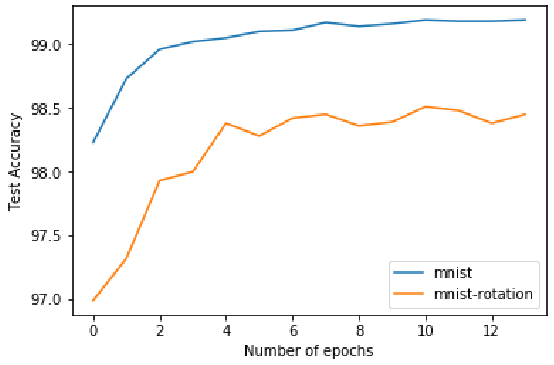
For MNIST classification task, the training loss and test accuracy plots are shown in Figure 8 and the maximum test accuracy is written in TABLE I. Self-supervision loss (via rotation task) does not seem to be improving the performance for MNIST classification but rather degrades the performance of the network. We posit that by adding the rotation task, we have increased the complexity. Also, since digits like 6 and 9, 2 and 5 are 180° rotations of each other, our method has introduced noise in the dataset. Hence, the classification accuracy for these classes has decreased which can be seen in TABLE II. The accuracy for classes 2, 5, 6, and 9 are worse for the model with self-supervised training via rotation task.

The results for the experiments over DeepFake dataset are given in TABLE III. We find that AlexNet [13] backbone trained with rotation loss performs poorly than the AlexNet backbone trained without it. This hints that the rotation auxiliary task does not lead to learning of good feature representations that generalize better for an out-of-distribution dataset.

For the BigBiGAN network, we obtain a particularly interesting result. For both random and hard splits, BigBiGAN with RevNet-50X4 [15] Encoder obtains better performance than AlexNet [13] and ResNet-50 [12] networks which have been trained in a fully supervised manner. Given that the BigBiGAN network has been trained on ImageNet without using any label information, the superior performance of the features obtained from the encoder of the BigBiGAN



(a) Training Loss



(b) Test set accuracy

Fig. 8: Training loss and Test accuracy for MNIST Classification with and without self-supervision (Rotation Task).

Class	MNIST	MNIST-Rotation
0	99.8	99.39
1	99.65	99.56
2	**99.32**	**98.93**
3	99.6	98.61
4	98.88	98.68
5	**98.99**	**97.98**
6	**98.96**	**97.7**
7	99.12	97.67
8	98.97	98.25
9	**98.51**	**97.52**

TABLE II: Class-wise accuracy on MNIST test dataset

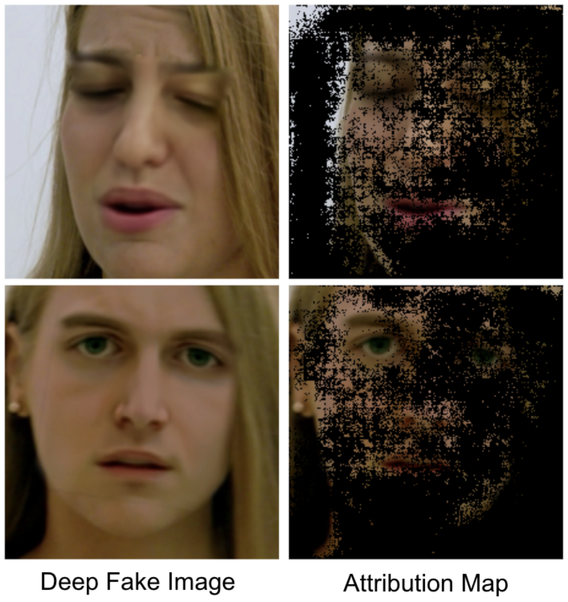


Fig. 9: Attribution Maps generated using Smooth-Grad [16], which computes map by performing squared average of the gradients with respect to the input for various noisy inputs. It can be seen that the model has learnt to identify relevant regions in the input image.

over those from networks utilizing label information highlights the importance of self-supervision. The abundance of unlabeled data in the web paves the way for such self-supervision representation learning using these unlabeled images to learn a good feature extractor. This feature extractor can then be used for any downstream task, even if the downstream task uses dataset that comes from a distribution different than the one used to train the feature extractor.

B. *Reinforcement Learning*

Figure 10 shows the plots of Mean reward with the number of training episodes for different methods in the game of Pong and Breakout. TABLE IV presents the best mean score in 100 episodes for different methods. We observe that the rotation prediction self-supervision task results in lower scores in both Pong and Breakout. Intuitively, this happens because the rotation task forces the network to look into non-significant areas of the game board like the score bar.

In case of sequence prediction, the score is almost double the baseline for Breakout while it is almost

Model	Random Split	Hard Split
ResNet-50 (Random Initialization)	56.7	56.2
ResNet-50 (ImageNet pre-trained)	96.1	91.1
AlexNet (ImageNet pre-trained)	81.5	73.5
AlexNet (Backbone trained with ImageNet & Rotation)	75.8	71.1
Classifier over BigBiGAN (ResNet-50 Enc.)	92.3	70.8
Classifier over BigBiGAN (RevNet-50X4 Enc.)	96.5	92.7

TABLE III: Binary Classification Accuracy for DeepFake Detection using different methods.

Agent	Pong	Breakout
Random	-21	1.25
DQN	20.0	9.9
Human [17]	-3	30.5
World Record [18]	21.0	864
Rotation	-20.1	7.6
Sequence	18.2	14.9

TABLE IV: Mean Score in 100 episodes for Pong and Breakout for various agents.

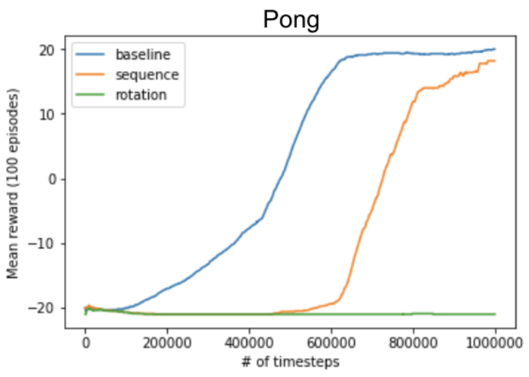
similar to the baseline for Pong. Hence, we observe that utilizing the dynamics of the game in the form of sequence prediction gives a stronger self-supervision objective.

Hence, we conclude that a self-supervision task for any Reinforcement Learning task should have the following properties:

1) The auxiliary self-supervision task should be complementary to the Q value prediction task so that mutual information in latent representation is shared between the two.
2) For better performance, the task should take into account the dynamics of the game rather than being just a function of the individual frames.

Further experiments using static image based and dynamic sequence based supervision task is required to firmly establish the above conclusions.

We also check whether self-supervision leads to learning of good feature representations. We plot the image attribution map corresponding to the sequence prediction branch after pre-training the network with only the self-supervised loss. To generate the attribution map, we use Smooth Grad [16], which computes map by performing squared average of the gradients with respect to the input for various noisy inputs. The attribution map is shown in Figure 11. It can be seen that the model has learnt to identify relevant regions in the input image using just the self-supervision loss.



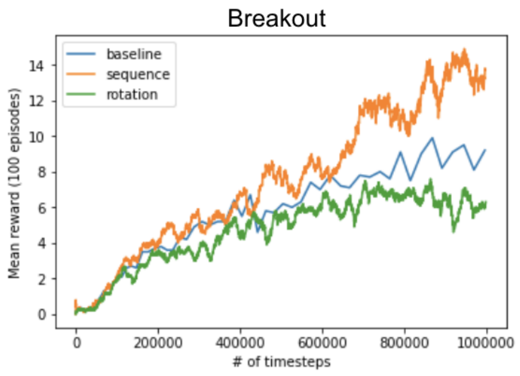
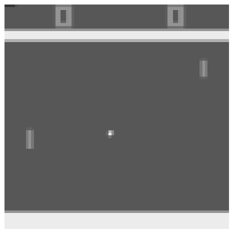
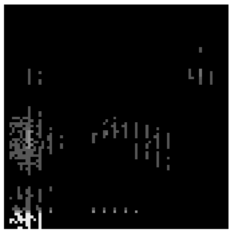


Fig. 10: Mean reward with number of episodes.



(a) Input frame



(b) Attribution map

Fig. 11: (a) Input state, and (b) the corresponding Image attribution Map. Here the network has been pre-trained with the self-supervised loss only and Q-learning is not performed. We can see that just with self-supervised training only, the network is able to learn useful features.

Layer 1: Convolution Input: 28x28; Output: 26x26x32 Kernel size: 3x3 No. of Filters: 64, stride: 1 Padding: SAME Activation: Relu
Layer 2: Convolution Input: 26x26x32; Output: 24x24x64 Kernel size: 3x3 No. of Filters: 64, stride: 1 Padding: SAME Activation: Relu
Layer 3: Pooling Max Pooling, Padding: SAME Kernel size: 2 Output: 12x12x64
Layer 4: Fully Connected layer Input: 9216; Output: 128 Activation: Relu
Layer 5a (Classification): Fully Connected layer Input: 128; Output: 10
Layer 5b (Rotation Prediction): Fully Connected layer Input: 128; Output: 4

TABLE V: Network architecture for MNIST Classification

VII. Conclusion

This project aims to explore the benefits of self-supervision in two classes of tasks, (a) Classification and (b) Reinforcement Learning. We augment standard supervised losses with those derived from self-supervised tasks to improve representation learning. We have currently explored (a) rotation angle prediction task and self-supervised representation learning via Bidirectional GANs in classification (detection of forgery in manipulated images), and (b) sequence-based self-supervision technique (i.e predicting the permutation) and rotation angle prediction in reinforcement learning. We observe that rotation prediction task does not seem to provide any help in both classes of tasks. In case of classification (DeepFake classification), self-supervised representation learning via BigBiGAN extracts useful features which can withstand domain shifts in datasets. Sequence-based self-supervision technique is complementary to the Q value prediction task and leads to an improved performance. Our work is a stepping stone for future research in the domain of using self-supervision in Reinforcement Learning and Classification.

References

[1] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[2] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European Conference on Computer Vision*, pages 577–593. Springer, 2016.

[3] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[4] Jong-Chyi Su, Subhransu Maji, and Bharath Hariharan. Boosting supervision with self-supervision for few-shot learning. *arXiv preprint arXiv:1906.07079*, 2019.

[5] Olivier J Hénaff, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

[6] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[7] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*, 2019.

[8] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[9] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

[10] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. *arXiv preprint arXiv:1912.01991*, 2019.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[14] Jeff Donahue and Karen Simonyan. Large scale adversarial

representation learning. In *Advances in Neural Information Processing Systems*, pages 10541–10551, 2019.

[15] Aidan N Gomez, Mengye Ren, Raquel Urtasun, and Roger B Grosse. The reversible residual network: Backpropagation without storing activations. In *Advances in neural information processing systems*, pages 2214–2224, 2017.

[16] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[17] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[18] Marin Toromanoff, Emilie Wirbel, and Fabien Moutarde. Is deep reinforcement learning really superhuman on atari? *arXiv preprint arXiv:1908.04683*, 2019.