Evaluating Learned State Representations for Atari

Adam Tupper
Department of Computer Science
and Software Engineering
University of Canterbury
Christchurch, New Zealand
adam.tupper@pg.canterbury.ac.nz

Kourosh Neshatian
Department of Computer Science
and Software Engineering
University of Canterbury
Christchurch, New Zealand
kourosh.neshatian@canterbury.ac.nz

***Abstract*—Deep reinforcement learning, the combination of deep learning and reinforcement learning, has enabled the training of agents that can solve complex tasks from visual inputs. However, these methods often require prohibitive amounts of computation to obtain successful results. To improve learning efficiency, there has been a renewed focus on separating state representation and policy learning. In this paper, we investigate the quality of state representations learned by different types of autoencoders, a popular class of neural networks used for representation learning. We assess not only the quality of the representations learned by undercomplete, variational, and disentangled variational autoencoders, but also how the quality of the learned representations is affected by changes in representation size. To accomplish this, we also present a new method for evaluating learned state representations for Atari games using the Atari Annotated RAM Interface. Our findings highlight differences in the quality of state representations learned by different types of autoencoders and their robustness to reduction in representation size. Our results also demonstrate the advantage of using more sophisticated evaluation methods over assessing reconstruction quality.**

I. Introduction

Deep reinforcement learning (RL) has enabled us to train agents directly from low-level observations of the environment, such as images [1]. However, improving the computational and sample efficiency of deep RL methods remains an open challenge. These constraints limit the applicability of using deep RL to solve real-world tasks, such as those in robotics, and largely confines deep RL to video games and tasks that can be modelled using simulations. To scale deep RL beyond environments that can be simulated, a renewed focus on state representation learning has emerged to enable more efficient downstream RL.

In the context of RL, state representations are compact descriptions of raw observations that preserve the important information needed for the agent to chose its actions [2]. State representation learning focuses on learning such state representations independent of learning a controller and without the supervision of the true state. Video games and simulations are useful tools for evaluating state representation learning methods because they allow easy access to the true state of the environment. This allows us to compare the learned state representations against the ground truth. For video games, important state variables are typically the locations of the agent and other objects, scores, and other game-specific information.

978-1-7281-8579-8/20/$31.00

Autoencoders are a popular class of neural networks used for state representation learning [3], [4]. They are trained to learn low-dimensional representations of observations through reconstruction, i.e. minimising the reconstruction error between original and reconstructed observations. However, reconstruction error is proxy measure for state representation quality that is assumed to, but may not necessarily, align with the desired goal to encode important state variables. For example, minimising reconstruction error does not ensure that the important state variables can be easily extracted from the learned representation. Nor does it ensure that small, yet potentially crucial details in the observation will be retained, since they contribute little to the overall reconstruction error.

In this paper, we investigate the true quality of the state representations learned by different types of autoencoders by assessing the quality of learnt state representations for a set of Atari games (one of the standard evaluation platforms for RL). For our investigations, we assess the quality of the encoding of important state variables for each game by employing a novel evaluation method that probes the contents of the learnt state representations using ground truth state information provided by the Atari Annotated RAM Interface [5]. Our evaluation method extends the original evaluation method proposed alongside this interface. Our results highlight the differences in the quality of state representations learned by different types of autoencoders, and also how the quality of the learned representations is affected by changes in representation size. Our results also demonstrate that the quality of reconstructions can deceive the quality of the underlying state representation, highlighting the need for sophisticated evaluation methods, such as the one we propose.

The remainder of this paper is organised as follows. Section II provides an overview of the different types of autoencoders assessed in this work and discusses different approaches for evaluating state representation learning methods. Section III describes our extensions to the approach for evaluating state representation learning methods proposed in [5]. Section IV describes the experimental setup used to evaluate the autoencoders. Section V presents the results of our experiments and discusses the findings. Finally, Section VI presents our conclusions.

II. Background and Related Work

This section provides a brief overview of representation learning through the use of autoencoders, followed by a discussion of different techniques for evaluating learned state representations. For a comprehensive review of research in state representation learning, we direct readers to [2].

A. Autoencoders

Autoencoders are a class of neural networks that are trained to learn low-dimensional representations of data. They consist of two halves: an encoder and a decoder. The encoder learns a function $f(\boldsymbol{x})$ that maps an input vector $\boldsymbol{x}$ to a latent space encoding or compressed representation $\boldsymbol{z}$. The decoder learns a function $g(\boldsymbol{z})$ that maps the encoding $\boldsymbol{z}$ back to a reconstruction of the input $\hat{\boldsymbol{x}}$. Setting the size of $\boldsymbol{z}$ less than the size of $\boldsymbol{x}$ introduces an *information bottleneck* which forces the network to learn to extract only the features of the data that are most important for reconstruction. Autoencoders that rely on this bottleneck alone to force the network to learn which aspects of the data are important are known as undercomplete autoencoders (AEs).

Variational autoencoders (VAEs) [6] use a statistical approach for learning compact encodings. They assume that the training data is drawn from a distribution that can be parameterised by a vector of latent variables $\boldsymbol{z}$. They attempt to learn a probability distribution for each latent variable, through a process called variational inference. This contrasts with undercomplete autoencoders that output a single value for each latent variable. When decoding, we sample from each distribution to generate a vector to serve as the input for the decoder. An advantage of this approach is that by learning a distribution for each latent variable, we force the encoder to learn a smooth, continuous latent space representation of the data, where similar observations should be located close to each other in the latent space. VAEs are trained to minimise a loss function consisting of the reconstruction error and the KL divergence.

Disentangled variational encoders (β-VAEs) [7] introduce a parameter $\beta > 1$ that assigns a higher weight to the KL divergence term in the loss function. Greater penalisation of the difference between the distributions places a larger emphasis on ensuring that each latent variable encodes a different attribute in the data. Prior work on the use of disentangled variational autoencoders in reinforcement learning environments has shown the benefits of state representations where each latent variable encodes a different property of the environment [3]. This may simplify the encoding and make policy learning easier.

B. Evaluating Learned State Representations

A brute force approach for evaluating different state representation learning methods is to train an agent to perform a particular task using the representations learned by each method. Although this approach has been commonly used in the past [4], it can be a costly and inefficient use of time and computation. Realistically, such an approach is only possible for simple tasks where agents can be trained within reasonable time. Furthermore, the choice of RL algorithm used for training the agents may bias the results. It has been regularly demonstrated that different types of algorithms, such as value-based methods like Deep Q-Learning [1] and policy gradient methods like Proximal Policy Optimisation [8], are effective for different tasks, even within the same class of problems, such as Atari games. A better approach to evaluating state representation learning methods is to devise a method of evaluation that is independent of the control algorithm applied for the downstream RL task.

For state representation learning methods that learn by reconstructing observations, one alternative to training agents is to visually compare the quality of the reconstructions against the original observations. However, as mentioned previously, reconstruction quality does not necessarily align with representation quality from a policy learning perspective. Furthermore, this approach can only be used to evaluate techniques that learn by reconstructing observations. A similar but more general approach is to visually compare the observations of nearest neighbours in the learned state space to see if they encoder similar observations [9], [10]. k-nearest neighbour mean squared error (KNN-MSE) [11] and normalisation independent embedding quality assessment (NIEQA) [12] are quantitative methods that perform more comprehensive evaluations in this manner. These remove the need for manual checking.

A final approach is to "probe" the learned representations by training small regression or classification models to predict ground-truth data from learned representations [5], [13]. Although this approach relies on access to ground truth data for each environment, it allows for a much more detailed evaluation of the information stored in the learned state representations. Anand et al. [5] used such a method for evaluating learned representations for Atari games using linear classification probes (single layer neural networks). Jonschkowski et al. [13] trained non-linear regression probes (neural networks with three hidden layers) to predict the positions and velocities of the cart and poles for pole balancing tasks.

Anand et al. [5] introduced the Atari Annotated RAM Interface (AtariARI) to enable the evaluation of state representation learning methods using Atari games. The AtariARI provides RAM annotations for 22 Atari games supported by the OpenAI Gym toolkit [14]. These annotations identify which of the 128 bytes of RAM store values related to information displayed on screen, such as the position of the player. Through a wrapper for the existing OpenAI Gym interfaces for each supported Atari game, the RAM values for each state variable are made available alongside the observation at each time step. They proposed evaluating the quality of state representations using the AtariARI by training a linear classifier (probe) for each state variable that predicts the value using the condensed state representations as input. The performance of the classifiers give an indication of the quality of the encoding of each variable. Furthermore, they assigned each state variable to one of five categories (agent localisation, object localisation, small object localisation, score/clock/lives, and miscellaneous)

to allow for comparisons between games by aggregating the results within categories.

III. Proposed Representation Evaluation Method

Our state representation evaluation method extends the probing method proposed by Anand et al. [5] by introducing regression and non-linear probing. In the subsections we describe our motivation and implementation for these extensions.

A. Regression Probes

Anand et al. [5] formulate the task of predicting each state variable as separate 256-way classification problem, as each byte of RAM can represent 256 possible values, regardless of the nature of each variable. While this makes aggregating the results over all state variables for each game easier, it ignores the ordinal nature of many of the variables. For instance, those that store the positions of objects along the x or y axis of the screen. When framing the tasks of predicting the values of these variables as a classification problem, an off-by-one error is equally as bad as an off-by-100 error. Given the importance of localising objects for learning to play the game, it is important that this information is evaluated as accurately as possible. To address this issue, we propose using regression probes for appropriate variables.

Examining the state variables for each of the games supported by the AtariARI, we found that approximately three quarters of all variables are numerical, and thus better suited to evaluation using regression rather than classification. Of these variables, the vast majority fall under the localisation categories. Furthermore, the localisation categories are the only categories in which all the state variables for all games are better suited for regression. Therefore, these are the only categories that we evaluate using regression probes. Although the majority of variables within the score/clock/lives category would also be better evaluated using regression, this is not the case for all games. For example, for games in which the scores achieved by the agent and/or opponent are low, such as Pong, the scores are stored using a single byte of RAM. However, for games where the agent can achieve a higher score than 255, such as Asteroids, the score is stored using multiple bytes of RAM. Because the score is split between bytes, we cannot use regression to evaluate these variables.

For regression, we require a similar metric to the $F1$ score that is used for classification that can (a) provide a good indication of predictive performance, and (b) be meaningfully averaged within and across categories. For this, we use the coefficient of determination, R^2, between the predicted and target values. The R^2 value provides a good indication of model fit and predictive power of the regression probes trained for each state variable. It is defined as follows:

$$R^2 = 1 - \frac{\sum(y - \hat{y})^2}{\sum(y - \bar{y})^2} \tag{1}$$

where y is the target value, $\hat{y}$ the predicted value, and $\bar{y}$ the mean of the target values. An R^2 value of one means that the model (in this case our probe) is able to perfectly predict the target values, whereas a value of zero indicates that the model is unable to make good predictions. In this way, performance is described similarly to the classification probes using accuracy or $F1$ scores.

B. Non-Linear Probes

Our second proposed change is to train non-linear probes instead of linear ones. The rationale behind this decision is that the information on state variables may be compressed in a non-linear manner that cannot be extracted using a linear classifier/regressor. This may be particularly true as the representation size is decreased and the compressors are forced to compress the state information into a smaller vector. While non-linear encoding introduces complexity, typical policy networks are sufficiently complex to learn from non-linear data. Therefore, we should allow the evaluation method to be flexible enough to account for this. We propose using a single hidden layer in the probes, equal in size to the representation (input layer) size as a good compromise between allowing for too much non-linearity and not assessing the true content of the representations. The nodes in the hidden layer used Rectified Linear Unit (ReLU) nonlinearities.

IV. Evaluating the Quality of State Representations Learned by Autoencoders

In this section, we describe our experimental setup and procedure for comparing the quality of the representations learned by different types of autoencoders with different representation sizes, using our evaluation method described in Section III.

A. Experimental Design

The goal of our experiment is to compare the quality of the representations learned by different types of autoencoders with varying representation sizes. We compare undercomplete (AE), variational (VAE), and disentangled variational (β-VAE) autoencoders. We train an autoencoder of each type with 10 different representation sizes in the range of 10 to 100 dimensions, in increments of 10 dimensions. The architecture, with the exception of the bottleneck, and training hyperparameters are held constant for all models. We evaluate each model on a set of four games: Asteroids, Boxing, Ms Pacman, and Pong. These games were chosen because they are (a) very different in appearance, (b) contain a wide range of objects of different shapes and sizes, and (c) represent a range of complexity in terms of their game states.

B. Autoencoder Architecture and Training

All of the autoencoders share the same encoder and decoder architectures. The only difference between the architecture of AE models and VAE and β-VAE models is the implementation of the bottleneck. The decoder is symmetrical to the encoder. The encoder architecture has five layers, all of which use 3×3 kernels. The first three layers learn 32 filters, while the last two learn 64 filters. The flattened output of the final convolutional layer is connected to a bottleneck of a particular size using a final fully connected layer.

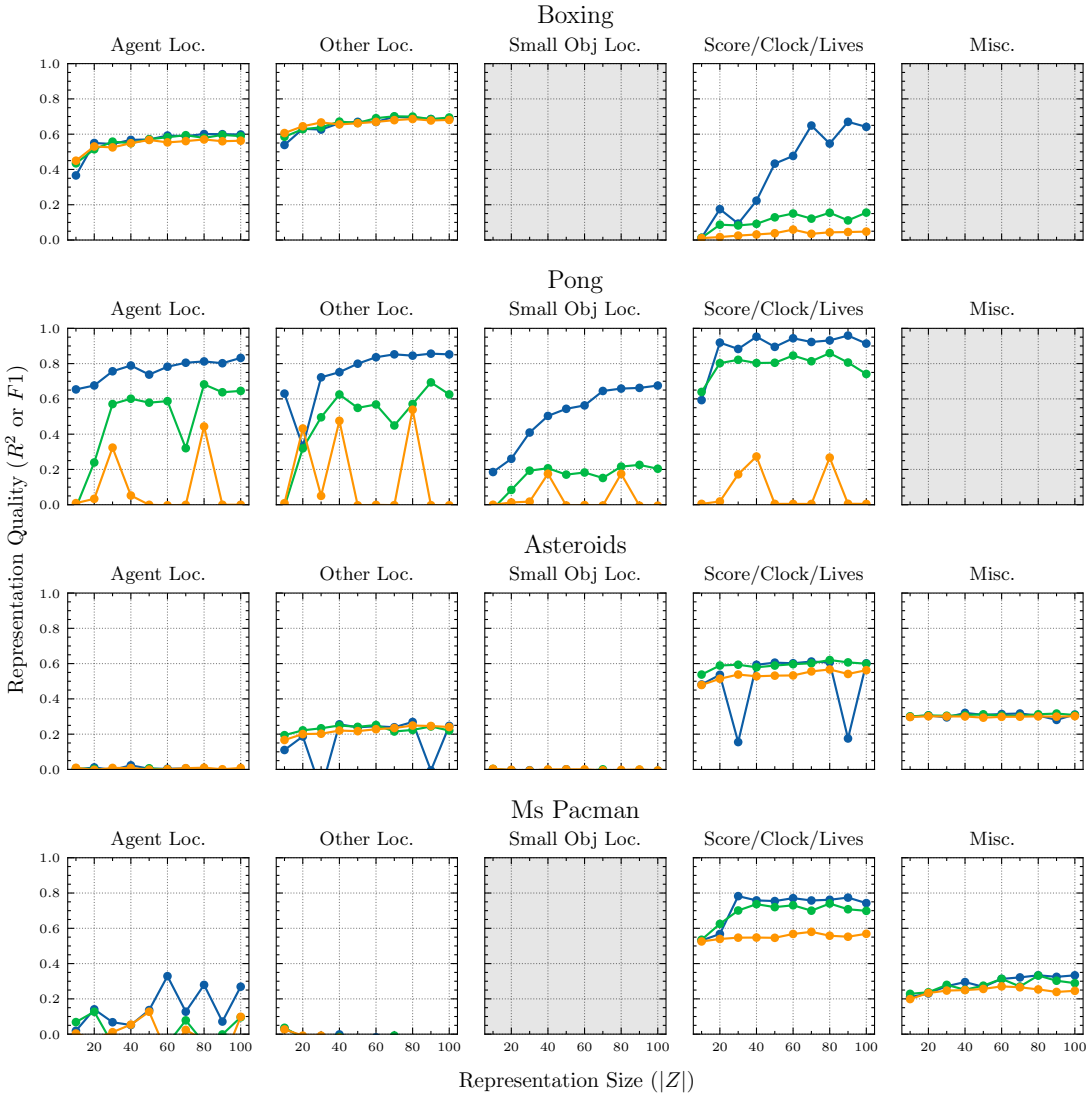

Fig. 1. The results of our AtariARI evaluations. Each colour depicts the results for a different type of autoencoder: ● AE, ● VAE, ● β-VAE. The localisation categories are measured using R^2, while the remainder are measured using $F1$ scores.

The learning rate ($1e^{-4}$), batch size (64), maximum number of training epochs (50), and the KL divergence weighting, β (4) for the β-VAE were chosen through informal experimentation. We used early stopping with a patience of 10 epochs to stop training early if performance plateaued. Each model was trained using a dataset of 110,000 unique, full-size (210×160 px), greyscale gameplay images for each game. These images were split into 100,000 training and 10,000 validation images. The images were collected by PPO agents trained using the implementation provided by the Stable Baselines reinforcement learning algorithm library [15]. Agents trained using this algorithm were chosen because they are relatively quick to train and have been shown to be very high performing when trained to play Atari games [8]. We trained these agents using the same hyperparameters used to obtain the original PPO results. We used trained agents because they are able to (a) collect a sufficient number of images, and (b) explore more of the state space than random agents. The autoencoders are trained to minimise the sum of squared errors (SSE) reconstruction error measure, in addition to the KL divergence in the case of the VAE and β-VAE models.

C. Probe Training

We trained the regression and classification probes following the same procedure outlined by Anand et al. [5]. For each game, we collected 45,000 unique frames and the values of the state variables provided by the AtariARI at the corresponding time step using the same trained PPO agents used to collect the images for training the autoencoders. These images were split into 35,000 training, 5,000 validation, and 10,000 test images. A probe was trained for each state variable for each game, using the Adam optimiser and a learning rate scheduler with an initial learning rate of $5e^{-4}$. Each time the validation plateaued for five epochs on the validation set, the learning

rate was decreased by a factor of 0.2, to a minimum of $1e^{-5}$. Each probe was trained for a maximum of 100 epochs, but with early stopping if the validation loss plateaued for 15 epochs. Classification probes used cross-entropy loss, whereas regression probes used mean squared error loss. Following training, each model was evaluated on the test set.

V. Results

This section presents the results of our evaluations of the state representations learned by different autoencoders.

A. AtariARI Evaluation Results

Fig. 1 presents the results of our evaluations of the quality of the representations learned by different types of autoencoders with different representation sizes.

We observed the most interesting patterns for Boxing and Pong, the games with lowest number of state variables. For Boxing, we observed that all types of autoencoders performed similarly when learning positions of the agent and opponent, with little degradation in representation quality as size decreased. However, all types of autoencoders showed a steep decline in performance (most prominent for the position of the agent) as representation size decreased from 20 to 10 dimensions. When learning the value of the clock, and the player's and opponent's scores, the undercomplete autoencoder did a far better job than the variational autoencoders, although performance declined steadily as the representation size decreased. The VAE outperformed the β-VAE consistently.

For Pong, we observed much more pronounced differences in performance between the different types of autoencoders, and also different trends in performance as representation size decreased. Once again, the undercomplete autoencoder performed better than the VAE, and the VAE better than the β-VAE. For agent and opponent (paddle) localisation, there was little decline in performance until the representation size was reduced to less than 40 dimensions. From that point onward, the VAE performance dropped drastically, while the AE performance more moderately. For ball localisation, we observed a similar result to the *Score/Clock/Lives* category in Boxing. For the *Score/Clock/Lives* category in Pong, performance for both the AE and VAE models remained very high until the representation size was decreased to 10 dimensions.

For Asteroids and Ms Pacman, games with far greater numbers of state variables and more visually complex observations, all autoencoders performed far worse across the categories of state variables. There was little consistent difference in performance between the types of models. Performance was best for both games in the *Score/Clock/Lives* category. For Ms. Pacman, this was the only category in which there were clear differences in the quality of representations between models.

The comparatively poor performance for all models in Asteroids and Ms Pacman compared to Boxing and Pong suggests that the increased complexity of the games hampered performance across all categories of state variables. Although AE models performed best overall, a somewhat surprising result given the potential advantages offered by the VAE and β-VAE models, the variation in results between games makes it difficult to identify categories that are particularly well suited to any one architecture. Perhaps general trends would emerge if the evaluations were performed for a larger set of games.

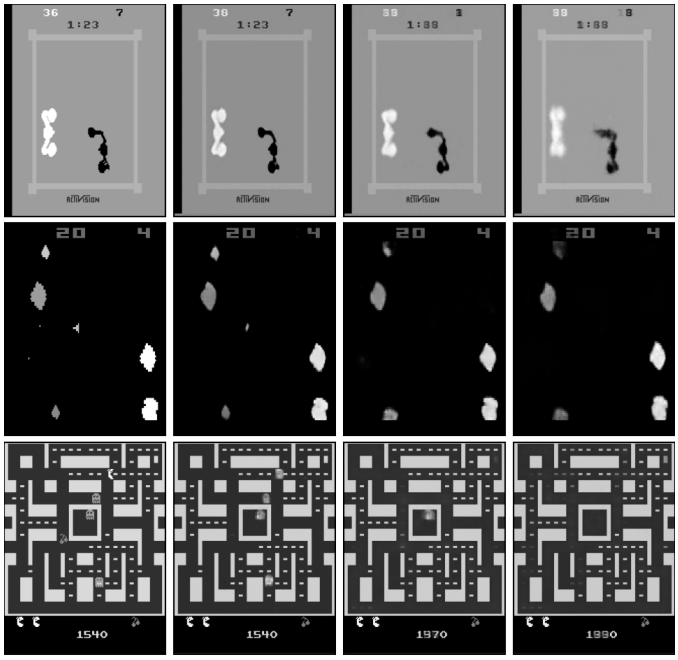

Fig. 2. Original and reconstructed images for Boxing, Asteroids, and Ms Pacman using different autoencoders, all learning 100-dimension representations.

B. AtariARI Evaluations vs. Reconstructions

One of the most interesting results we observed was how deceptive the reconstruction quality can be. In each game, there were significant discrepancies between the AtariARI results and the perceived quality of the representations based on the reconstructions.

One of the best examples of this was the quality of the reconstructions of the agent and opponent in Boxing (shown in Fig. 2), compared to the middling AtariARI performance. We observed that while all models were able to accurately reproduce the positions of the agent and opponent, the R^2 value for the regression performance topped out at approximately 0.6. One plausible reason for this is that the information, while clearly present in the compressed representations, was too highly compressed even for the non-linear probe to extract and process to a higher level. Furthermore, the AtariARI evaluation also shed greater light on situations where performance appeared poor or indifferent between models based on reconstructions alone. For example, the reconstructions of the scores and clock in Boxing appeared equally poor between variational and disentangled variational autoencoders, however, the values were far the better encoded by the variational autoencoders.

For Pong, despite the reconstruction of the ball remaining relatively clear and accurate for all undercomplete autoencoders, even when learning a representation of just 10 dimen-

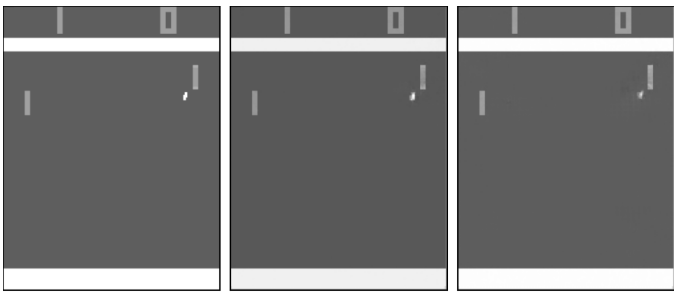

Fig. 3. Original and reconstructed images for Pong using different undercomplete autoencoders (AEs). $|Z|$ denotes the size of the learnt representations.

sions, as shown in Fig. 3, the AtariARI performance dropped drastically as representation size was reduced. In addition, the drastic drop in performance exhibited in the AtariARI results for the *score/clock/lives* category when the representation size dropped from 20 to 10 dimensions was not accompanied by a noticeable drop in reconstruction quality.

For Asteroids, despite the fact that all models were able to reproduce the positions and shapes of large asteroids, as shown in Fig. 2, the AtariARI localisation results were poor. Furthermore, although the undercomplete autoencoder with a representation size of 100 dimensions was able to reproduce the position of the agent and the positions and shapes of both large and small asteroids, unlike the VAE and β-VAE models, this is not reflected in the AtariARI results.

Finally, for Ms Pacman we again noticed that despite better reconstructions (being able to consistently reconstruct the positions of the agent and ghosts), the undercomplete autoencoder did not offer substantially better representations than the VAE and β-VAE autoencoders when more comprehensively evaluated using the AtariARI.

Overall, the results demonstrate that the AtariARI evaluations provide a far more detailed breakdown of performance than reconstructions alone. They also identified drops in representation quality that would have gone otherwise unnoticed if only the quality of the reconstructions were considered.

VI. Conclusions

In this paper, we investigated the quality of state representations learned by undercomplete, variational and disentangled variational autoencoders for a set of Atari games. To evaluate the quality of the learned representations we proposed and utilised novel extensions to an evaluation method that probes the representations using the AtariARI [5]. Our results demonstrated the differences in representations learned by different types of autoencoders and how representation size affects the quality of representations for each type. Our results also highlighted discrepancies between reconstruction quality and the quality of the encoding of important state variables in the learned representations, which illustrates the need for more thorough evaluation methods, such as the one we proposed. Overall, this work provides the most comprehensive evaluation yet of the use of autoencoders for state representation learning.

A. Future Work

For future work, we have identified several promising avenues of improvement and investigation:

- Investigating *why* the differences in performance between types of autoencoders exist.
- Comparing the performance of autoencoders against other state representation learning techniques.
- Investigations using a wider set of games, and across domains other than Atari games to build a holistic view of the ability to learn state representations using autoencoders across a wide and varied range of environments.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, p. 529, Feb. 2015.

[2] T. Lesort, N. Díaz-Rodríguez, J.-F. Goudou, and D. Filliat, "State representation learning for control: An overview," *Neural Networks*, vol. 108, pp. 379–392, Dec. 2018.

[3] I. Higgins, A. Pal, A. A. Rusu, L. Matthey, C. Burgess, A. Pritzel, M. Botvinick, C. Blundell, and A. Lerchner, "DARLA: Improving Zero-Shot Transfer in Reinforcement Learning." in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 1480–1490.

[4] H. van Hoof, N. Chen, M. Karl, P. van der Smagt, and J. Peters, "Stable reinforcement learning with autoencoders for tactile and visual data," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2016, pp. 3928–3934.

[5] A. Anand, E. Racah, S. Ozair, Y. Bengio, M.-A. Côté, and R. D. Hjelm, "Unsupervised State Representation Learning in Atari," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., Dec. 2019, pp. 8766–8779.

[6] D. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv*, vol. abs/1312.6114, Dec. 2013.

[7] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "Beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework." in *5th International Conference on Learning Representations, Conference Track Proceedings*, 2017.

[8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv*, vol. abs/1707.06347, Jul. 2017.

[9] L. Pinto, D. Gandhi, Y. Han, Y.-L. Park, and A. Gupta, "The Curious Robot: Learning Visual Representations via Physical Interactions," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 3–18.

[10] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain, "Time-Contrastive Networks: Self-Supervised Learning from Video," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 1134–1141.

[11] T. Lesort, M. Seurin, X. Li, N. Díaz-Rodríguez, and D. Filliat, "Deep unsupervised state representation learning with robotic priors: A robustness analysis," in *2019 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2019, pp. 1–8.

[12] P. Zhang, Y. Ren, and B. Zhang, "A new embedding quality assessment method for manifold learning," *Neurocomputing*, vol. 97, pp. 251–266, Nov. 2012.

[13] R. Jonschkowski, R. Hafner, J. Scholz, and M. Riedmiller, "PVEs: Position-velocity encoders for unsupervised learning of structured state representations," in *New Frontiers for Deep Learning in Robotics Workshop at RSS*, 2017.

[14] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI Gym," *arXiv*, vol. abs/1606.01540, Jun. 2016.

[15] A. Hill, A. Raffin, M. Ernestus, A. Gleave, A. Kanervisto, R. Traore, P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, and Y. Wu, "Stable Baselines," *GitHub repository*, 2018.