Technical report:

Experimenting with Autoencoders to Encode World State Representations

Blaž Meden

September 16, 2019

Abstract

In this work we implemented a small framework for training and testing auto-encoding architectures for the purpose of encoding world state representations into latent space and thus efficiently compressing the essential world properties depicted in the training image dataset. We explored the possibilities of incorporating a forward model in between encoder and decoder, to predict next state in the series of temporal image sequence, to see if forward model is able to produce meaningful representations (even if they are not interpretable by visual inspection).

1 Introduction

Autoencoders have the potential to be useful in the problem of representation learning, due to their specific architecture, which introduces a dimensional bottleneck in the central part of the model and by doing that, compresses the representation of the information in the original data. Training is usually done in a non-supervised way and it is very simple to do, as we are trying to learn how to reconstruct the original input image on the output via backpropagation. The only condition that is introduced with the topology of the architecture is the compression stage, which is encoding the hidden representations (better known as a latent space).

A typical autoencoder model is following the encoding-decoding architectural scheme. The encoder as frontal part of the model performs downsampling of the image and produces latent representation. Then, decoder takes the latent representation and tries to reconstruct the original input, based on given latent information. The model itself is trained in an end-to-end fashion and by optimizing entire model via backpropagation we minimize the loss between original inputs and obtained outputs. The loss is usually defined as metric for comparing two images (i.e. MSE). Other losses can be used as well. Other specific topological parameters can be also defined within the model, for example number of convolutional layers, size and number of filters per each convolutional layer, the size of the latent space and the activation functions, used after each layer (among other, more advanced options, such as regularization and initialization, etc.).

2 Related Work

Related area in our line of work can be divided into techniques for obtaining compressed representations (representation learning) and autoencoding deep learning architectures, which are one specific way of achieving latent representations.

2.1 Autoencoders

The idea of autoencoders (AE) evolved over time, although one of the first mentioned ideas of using a bottleneck inside neural network was in Rumelhart's work [1]. As of nowadays, a well written and detailed chapter on the topic of autoencoders is available in the Deep Learning book from Goodfellow et al. [2]. The topic of autoencoders is becoming more diverse day by day, as researchers are using these kind of networks to do denoising, representation learning, etc.

Variational Autoencoder (VAE), as an extension of AE was introduced by Kingma et al. [3]. The paper includes formal definitions and mathematical foundations of the approach as well as visual results from two datasets (Frey Faces and MNIST digits).

2.2 State Representation Learning

In the World Models by Ha [4], a VAE is used to learn about state representations in a game-like environments. In this work, a VAE model is successfully used to reconstruct observations and provide latent space, which encodes crucial scenario information needed to train the gameplay agents.

Lesort et al. [5] on the other hand, surveys other approaches for learning state representations. Multiple approaches and ideas are discussed through the paper, namely the forward and inverse modelling, reconstruction of the observation, prediction of next observation and usage of actions, state constraints and rewards during the learning process.

3 From Images to Latent Space Representation

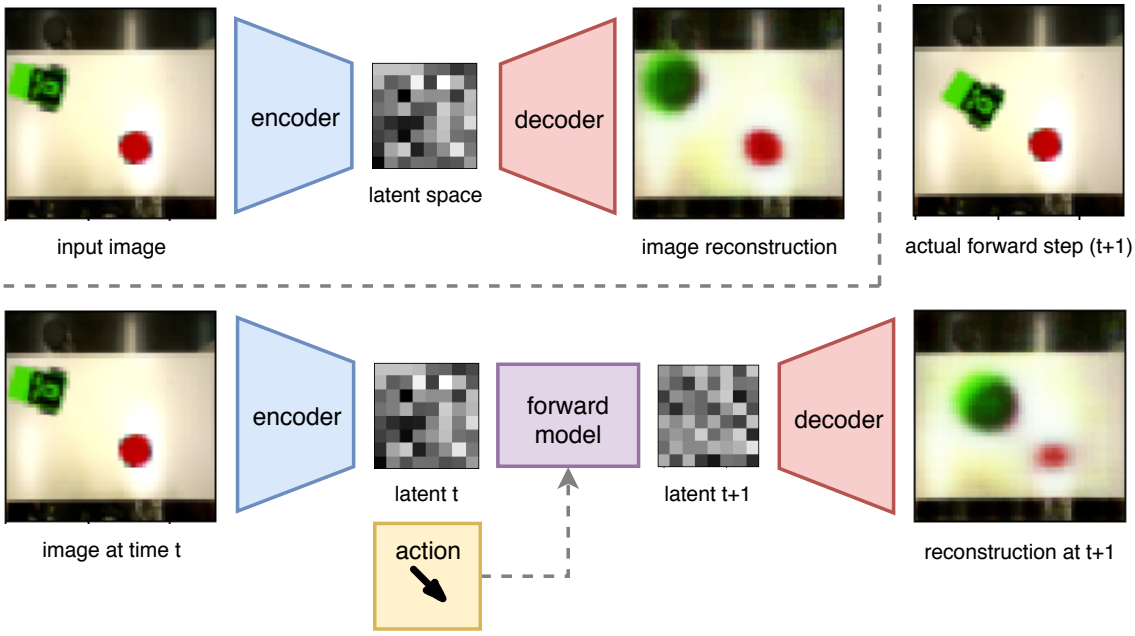


Figure 1: Pipeline overview of our experiments. Top marked section illustrates first tier of experiments, where we evaluated the reconstruction quality of images from our scenario. Section at the bottom illustrates the second part of the experiments, where we included separately trained forward model to predict latent representation, which can be used to reconstruct image at next simulation step. Ground truth image is also available in top right corner of the figure, illustrating that reconstruction is closer to next simulated step than the original input image.

In our case, we were experimenting with two variations of autoencoders. A regular, fully convolutional autoencoder (AE) and a variational autoencoder (VAE). The main difference between these two models is the fact, that AE is essentially only a series of layered convolutions, applied on the input image, to produce a final result. In its most simplest form, this could be represented with simple interpolation (downsampling, followed by upsampling), but in the case of AE, the filter parameters are learned (so downsampling and upsampling operations are tuned during the training, to minimize output loss). VAE on the other hand, uses the same logic, however, in between the encoder and decoder, there is a special sampling layer, which is fitting the training data onto randomly sampled variables from multivariant Gaussian distribution. This essentially scrambles the spatial information, but on the other hand, dedicates entire latent space to encode the hidden representation. In this sense, the variational latent space seems more rich with information, but the encoding cannot be visually interpreted.

3.1 Reconstruction with Autoencoders

Our work includes two series of experiments. First experiment was intended to study whether the information in the latent space can be reconstructed back into the original images. We tried

multiple configurations of models (testing AE and VAE, and also changing the dimensions of latent spaces, attention factors of the foreground and background in the training images, object sizes). At the end of the experiments we obtained best possible AE and VAE architectures, which were then used in the second part of the experiments.

3.2 Forward Model Experiments

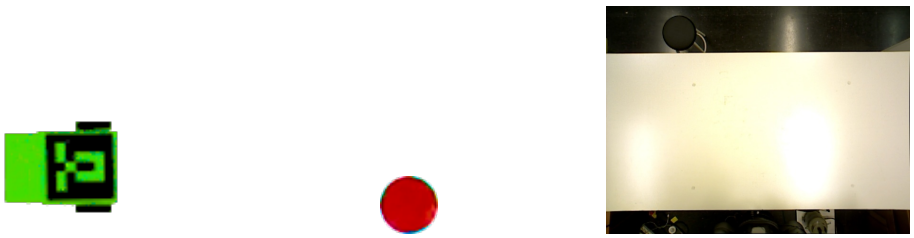
Second part of the experiments included introduction of forward model in between encoding and decoding part of the AE/VAE. In this case, we defined an action, which denotes the angle in which the robot move in each time step. The action and latent space, obtained from the encoder represented the input into our forward model. The expected output of the model is supposed to be the latent space in the next time step (essentially predicting, what would happen if the action at time t was executed, knowing the latent space at time t – meaning that the model needs to reconstruct the scene at time $t + 1$). Since we encountered some issues with model convergence, while training it on the original data, we here report the results on simplified version of the data (segmentation-like looking grayscale sequences) without color-rich information within the training images. See Figure 1 for pipeline overview.

4 Results

In this section we describe the generation of training and testing data and the results obtained during first and second part of the experiments (image reconstruction and forward model state prediction).

4.1 Dataset

Dataset images were generated using the images displayed in Figure 2. We simulated a simple robot on a plane. The task of the robot was to find the relevant red circle on the table. For the first set of experiments, the position of the robot and the circle was placed randomly on the bright part of the background image (table area). To simulate image sequences in the second step of experiments we implemented simple, brownian-like exploration in which the robot moved around the table to explore its environment. The probability of the robot moving around in a random pattern was 1/2. We also added a direct movement of the robot to the red target with a probability of 1/2, to make it converge faster to the goal. With the color images we also constructed corresponding ground truth segmentation masks, where foreground objects (the robot and the circle) were separated from the background. The masks, visible in Figure 3, were later used to calculate performance metric in form of Intersection over Union (IoU) on reconstructed images.



(a) (b) (c)

Figure 2: Building blocks of our generated dataset: a) a robot, b) an object of interest, and c) the background image.

4.2 Reconstruction Experiment

To conduct our first experiment, we trained various combinations of parameters, defining autoencoder architectures and training parameters. Common training parameters include Adam optimizer with learning rate $lr = 0.001$ with weighted mean squared error loss function (using weighted segmentation masks as balancing factors for reconstruction of the foreground objects and the background), batch

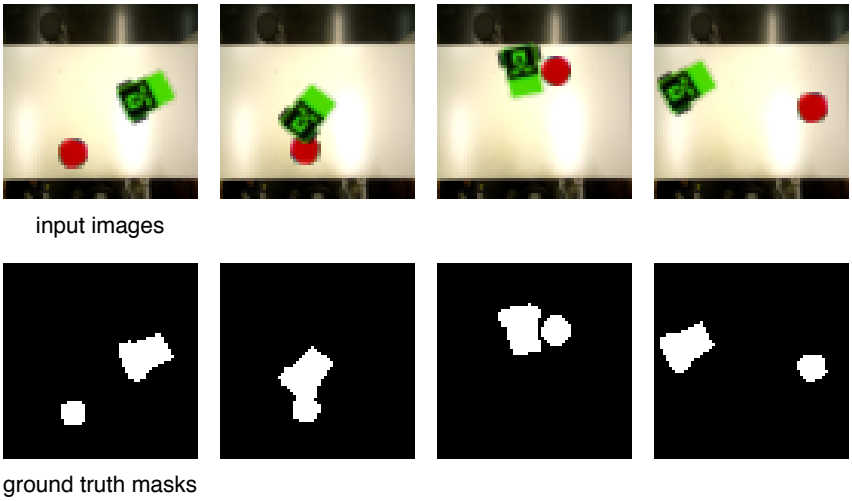


Figure 3: Generated image examples (top) and corresponding ground truth masks (bottom).

Model	Latent size	Back att.	Obj. att.	Obj. size	IoU (avg)	IoU (stdev)
VAE	64	0.2	0.8	3	0.65520	0.07533
VAE	64	0.33	0.67	3	0.61377	0.07079
AE	16	0.2	0.8	3	0.57823	0.07041
VAE	64	0.2	0.8	2	0.57033	0.11629
AE	64	0.2	0.8	1	0.56386	0.15444
VAE	64	0.33	0.67	2	0.52029	0.11642
VAE	16	0.2	0.8	3	0.50127	0.07992
AE	64	0.2	0.8	3	0.50019	0.05898
VAE	64	0.02	0.98	1	0.47663	0.06301
VAE	16	0.33	0.67	3	0.46870	0.08014

Table 1: Top 10 rated AE / VAE models in terms of average Intersection over Union (IoU) during tests. Best results are achieved with larger latent space sizes and by increasing the object attention factor to 0.8 (denoting that robot and the object are more important for reconstruction than the background, however the background still has to be reconstructed to some degree). Models performed best, when object size was increased 3 times their original size (indicating that object size matters, when trying to reconstruct the scene as accurately as possible, as smaller regions tend to be discarded during encoding, which can lead to disappearance of smaller objects).

size of 32 and 50 training epochs (with 5000 images per epoch). As we were training multiple variations of autoencoders, we were changing the sizes of latent space (most commonly using 4, 16, 64 with AE and 2, 4, 9, 16, 64 with VAE, see Figure 4). The dimension of the training images was $64 \times 64 \times 3$ (3-channel RGB images).

After the training we evaluated the final reconstructions with a test set of 64 images. An example of input images, latent spaces, reconstructed images and evaluation masks is available in Figure 4, where AE and VAE models were constructed using 16 and 64-dimensional latent spaces. Here we can observe, how latent space is represented in each model. AE keeps the spatial information and uses downsampled image version in order to reconstruct the final image (with higher dimensions producing better visual results). In contrast with AE, VAE uses entire latent space, to encode the representation, while scrambling the spatial information (encoded information is therefore richer but we cannot really interpret it). In the last row of Figure 4 we can also see the comparison with ground truth segmentation masks (green pixels denoting proper color classification, while blue and red represent misclassified pixels). Visual comparison of reconstructed images considering different latent space sizes is available in Figure 5. We also present the quantitative analysis in Table 1, where we report top 10 models with corresponding parameter combinations, which produced best accuracy in terms of IoU. In the table it is clear that larger object sizes and larger latent sizes are correlated with the overall accuracy of the models. Another important parameter is denoted as

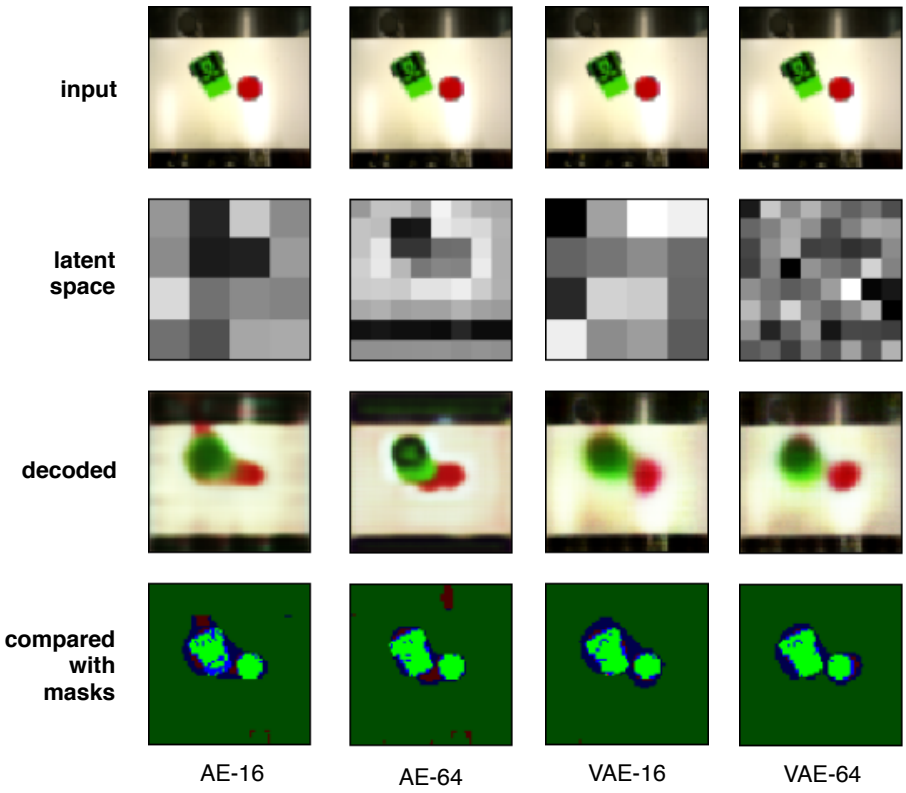


Figure 4: Comparison between AE and VAE with 16 and 64 latent space sizes. AE yields interpretable latent spaces while VAE manages to use all available latent variables to encode information resulting in scrambled-like looking representation.

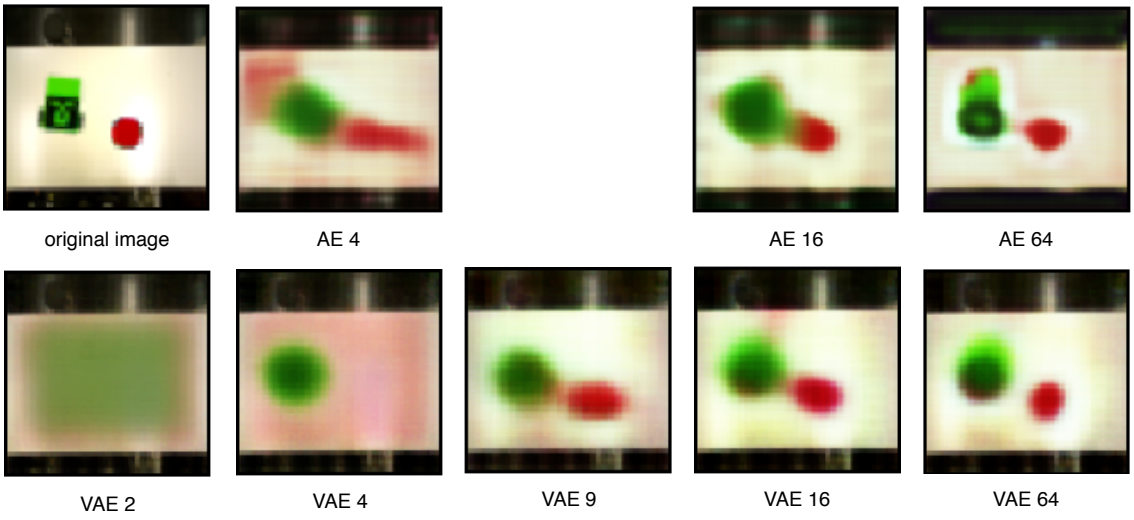


Figure 5: Qualitative reconstruction results from first part of the experiments. In left top corner an original reference image is shown. Top row illustrates image reconstructions with different sizes of latent spaces using regular convolutional autoencoder (namely 4, 16 and 64; latent spaces with sizes 2 and 9 are ommited for regular AE due to the topological constraints of the network). Bottom row similarly illustrates image reconstructions using variational autoencoder with different latent spaces (2, 4, 9, 16 and 64).

attention factor, which is in the best cases usually around 80% for foreground objects and 20% for the background. These results formed the basis for our selection of the models in the second part of the experiments.

Model	Diff. learning	Latent size	Obj. size	IoU (avg)	IoU (stdev)
VAE + FM	✓	64	3	0.35866	0.21140
VAE + FM	✗	64	3	0.35050	0.20697
VAE + FM	✓	16	3	0.19372	0.12674
VAE + FM	✗	16	3	0.18675	0.14988
AE + FM	✓	64	3	0.16381	0.12321
AE + FM	✗	64	3	0.14849	0.12366
AE + FM	✓	16	3	0.12763	0.10202
AE + FM	✗	16	3	0.05069	0.05329

Table 2: Results of second experiment in terms of average Intersection over Union (IoU). Various configurations were tested. VAE and AE were used to produce latent representations of the images and FM represents the forward model that was used to predict state in the next timestep. We tested latent sizes 64 and 16 with both autoencoders.

4.3 Forward Model Experiment

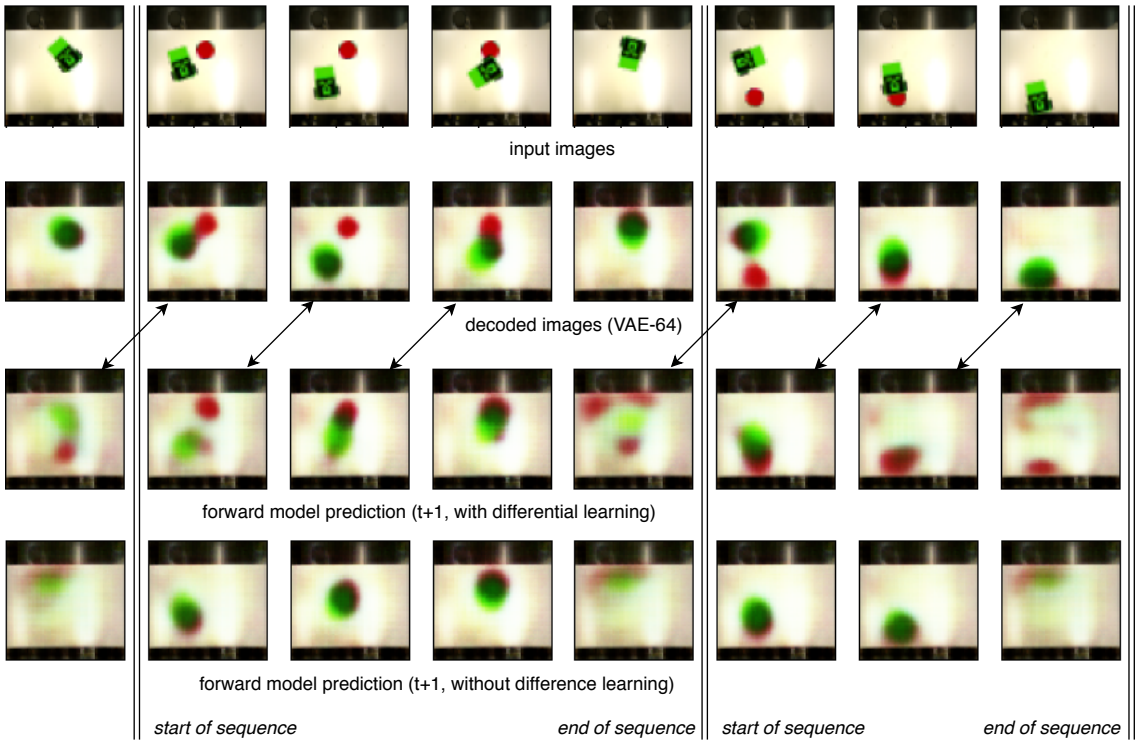


Figure 6: Prediction example using forward model (FM). Top row displays original sequence images in. Row in the middle represents corresponding reconstructed images in each timestep. Bottom row represents reconstructed images from latent space predicted by forward model, given the action of the robot.

In this part we used best pretrained autoencoder models (AE-64, VAE-64, AE-16, VAE-16). Forward model (FM) was trained separately on generated image sequences. The protocol of training FM included generating color images with the moving robot and red circle. Then we used pretrained encoder from autoencoder model to predict latent space, corresponding to each image in the generated sequence. We also generated a set of actions, each action denoting movement transition in between two consecutive image frames. The action was defined as the global orientation angle of the robot, normalized in between -1 and 1 (see the bottom row of Figure 1). Finally we trained FM with 50 epochs and 100 iterations per epoch with batch size of 32 (resulting in total of 160.000 generated images and latent spaces used for training). We also tried to train the forward model in two distinct ways. The default way was to use the model to predict next state, based on the current state of latent space and the action (passed into the model as a tensor of the same size

as the latent space, with repeated action values, to balance the network inputs). The second way of training was to predict only the difference between latent states in time t and $t+1$ (so-called difference learning). In the latter, the model only needs to predict the difference between the states as opposed to predict the complete state $t+1$ in the first variation. Results are gathered in Table 2. We can see that the difference between the proposed two ways of training the FM was not that obvious. The same observation can be made by analyzing the visual results in Figure 6. Otherwise we can notice, that the final image reconstruction accuracy decreases after we introduce the FM, which makes sense, since FM introduces new error within its predictions. However, some predictions are quite accurate (usually in between the sequence). Forward model of course fails to predict the states at the end of the image sequence, as the robot and the red circle get randomly replaced, after the robot reaches the circle (which indicates the end of the sequence). This behaviour is illustrated in Figure 6.

5 Conclusions

We tested different parameters with AE and VAE. Larger latent spaces yielded best results. Having larger object sizes also helped quite a lot. VAE-64 performed best overall. VAE uses entire latent space to encode non-interpretable information, while AE performs downsample and upsample, using only the spatial area around moving robot (interpretable latent space). Forward model is able to predict the movement of the robot to some degree. Reconstruction step and forward model both introduce some uncertainty and errors to final reconstruction. Forward model cannot predict unknown robot state after the sequence is finished, since the next robot initialization is random. This probably impacts the quantitative results the most, although there are failure cases, where the next action and next image are within the same sequence.

References

[1] D. E. Rumelhart and J. L. McClelland, *Learning Internal Representations by Error Propagation*. MITP, 1987. [Online]. Available: https://ieeexplore.ieee.org/document/6302929

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[3] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *CoRR*, vol. abs/1312.6114, 2013. [Online]. Available: http://arxiv.org/abs/1312.6114

[4] D. Ha and J. Schmidhuber, "World models," *CoRR*, vol. abs/1803.10122, 2018. [Online]. Available: http://arxiv.org/abs/1803.10122

[5] T. Lesort, N. D. Rodríguez, J. Goudou, and D. Filliat, "State representation learning for control: An overview," *CoRR*, vol. abs/1802.04181, 2018. [Online]. Available: http://arxiv.org/abs/1802.04181