
Accounting for the Sequential Nature of States to Learn Features for Reinforcement Learning

Nathan Michlo, Devon Jarvis, Richard Klein, Steven James
School of Computer Science and Applied Mathematics
University of the Witwatersrand
Johannesburg, South Africa
{nathan.michlo1, devon.jarvis1}@students.wits.ac.za
{richard.klein, steven.james}@wits.ac.za

Abstract

In this work, we investigate the properties of data that cause popular representation learning approaches to fail. In particular, we find that in environments where states do not significantly overlap, variational autoencoders (VAEs) fail to learn useful features. We demonstrate this failure in a simple gridworld domain, and then provide a solution in the form of metric learning. However, metric learning requires supervision in the form of a distance function, which is absent in reinforcement learning. To overcome this, we leverage the sequential nature of states in a replay buffer to approximate a distance metric and provide a weak supervision signal, under the assumption that temporally close states are also semantically similar. We modify a VAE with triplet loss and demonstrate that this approach is able to learn useful features for downstream tasks, without additional supervision, in environments where standard VAEs fail.

Keywords: representation learning, unsupervised learning, metric learning, reinforcement learning

1 Introduction

A fundamental challenge in machine learning is to discover useful representations from high-dimensional data, which can then be used to solve subsequent tasks effectively. Recently, deep learning approaches have showcased the ability of neural networks to extract meaningful features from high-dimensional inputs for tasks in both the supervised and reinforcement learning (RL) setting [1, 2]. While these learnt representations have obviated the need for manual feature selection or preprocessing pipelines, they are often not semantically meaningful, which can negatively impact downstream task performance [3]. Prior work has therefore argued that it is desirable to learn a representation that is *disentangled*. While there is no consensus on what constitutes a disentangled representation, it is generally agreed that such a representation should be factorised so that each latent variable corresponds to a single explanatory variable responsible for generating the data [4]. For example, a single image from a video game may be represented by latent variables governing the x and y positions of the player, enemies and collectable items.

In deep RL, there are two general approaches to feature learning. The first is to implicitly discover features by simply training an agent to solve a given task, relying on a neural network function approximator and stochastic gradient descent to uncover useful features. An alternative approach is to explicitly learn features through some auxiliary objective. One such method is to use variational autoencoders (VAEs) [5], which are trained on collected data to learn a lower-dimensional representation capable of reconstructing the given input. VAEs have been shown to produce disentangled representations when trained on synthetically generated data, although there is still no explicit reason why these representations should align with generative factors in the data.

In this work, we investigate the correspondence between the VAE reconstruction loss and learnt distances in the latent space. We demonstrate the ineffectiveness of state-of-the-art models by designing a simple visual gridworld domain over which VAE-based frameworks fail to learn useful features for a downstream RL task. We overcome this limitation by utilising the sequential nature of an RL replay buffer along with metric learning to guide representations learnt by VAEs. Furthermore, we design a disentangled metric learning approach which further improves learnt representations for downstream tasks.

2 Background

We model an agent acting in an environment by a Markov decision processes (MDP) $\mathcal{M} = \langle S, A, T, R, \gamma \rangle$ where (i) S is the state space; (ii) A is the set of actions; (iii) $T(s, a, s')$ describes the transition dynamics, specifying the probability of arriving in state s' after action a is executed from s ; (iv) $R(s, a)$ specifies the reward for executing action a in state s ; and (v) $\gamma \in [0, 1)$ is a discount factor. The agent’s aim is to compute a policy π that maps from states to actions and optimally solves the task. Given a policy π , RL algorithms often estimate a value function $v^\pi(s) = \mathbb{E}_\pi [\sum_{i=0}^{\infty} \gamma^i R(s_i, a_i) | s_0 = s]$, representing the expected return obtained following π from state s . The *optimal* policy π^* is the policy that obtains the greatest expected return at each state: $v^{\pi^*}(s) = v^*(s) = \max_\pi v^\pi(s)$ for all $s \in S$. A related quantity is the action-value function, $q^\pi(s, a)$, which defines the expected return obtained by executing a from s , and thereafter following π . Similarly, the optimal action-value function is given by $q^*(s, a) = \max_\pi q^\pi(s, a)$ for all states s and actions a [6].

For high-dimensional or continuous state spaces, representing the value function precisely is intractable. Instead, the value function is normally approximated through a parameterised function $v^\pi(s) \approx \hat{v}(s; \mathbf{w})$, where \mathbf{w} is some parameter vector to be learned; in deep RL, these parameters are the weights of a neural network that can be trained to approximate the value function. Since these neural networks require significant amounts of data to train, one way to improve sample efficiency is to use a *replay buffer*, where old experience data is stored in memory and reused to update the network as it continues to interact with the environment.

2.1 Representation Learning

Assume a dataset $\mathcal{X} = \{\mathbf{x}^{(0)}, \dots, \mathbf{x}^{(n)}\}$ is a set of independent and identically distributed (i.i.d)¹ observations $\mathbf{x} \in \mathbb{R}^N$, generated by some random process involving an unobserved random variable $\mathbf{z} \in \mathbb{R}^D$ of lower dimensionality $D \ll N$. Additionally, the true *prior distribution* $\mathbf{z} \sim p_*(\mathbf{z})$ and true *conditional distribution* $\mathbf{x} \sim p_*(\mathbf{x}|\mathbf{z})$ are unknown. Variational autoencoders (VAEs) aim to learn this generative process. Unlike autoencoders (AEs), which consist of an encoder $f_\phi(\mathbf{x}) = \mathbf{z}$ and decoder $g_\theta(\mathbf{z}) = \hat{\mathbf{x}}$ with weights ϕ and θ , VAEs instead construct a probabilistic encoder by using the output from the encoder or inference model to parameterise approximate posterior distributions $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$. The approximate posterior is then sampled from during training to obtain representations \mathbf{z} , which are then decoded using the generative model to obtain reconstructions $\hat{\mathbf{x}} \sim p_\theta(\mathbf{x}|\mathbf{z})$.

A *factorised Gaussian encoder* [5] is commonly used to model the posterior using a multivariate Gaussian distribution with diagonal covariance $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\sigma}_\phi(\mathbf{x}))$, with the prior given by the multivariate normal distribution

¹In RL, a replay buffer is not i.i.d, as data is generated by a sequential decision process. This can be mitigated via random sampling.

$p_{\theta}(z) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, with a mean of $\mathbf{0}$ and diagonal covariance \mathbf{I} . To enable backpropagation, the reparameterisation trick in Equation (1) is used to sample from the posterior distribution during training by offsetting the distribution means by scaled noise values.

$$\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}); \quad z = \mu_{\phi}(x) + \sigma_{\phi}(x) \odot \epsilon \quad (1)$$

VAEs maximise the evidence lower bound (ELBO) by minimising the loss given by Equation (2). VAE-based approaches often make slight modifications to this loss but generally the terms can still be grouped into regularisation and reconstruction components. The regularisation term constrains the representations learnt by the encoder, while the reconstruction term improves the outputs of the decoder so that they better match the inputs to the encoder. These terms usually conflict in practice, strong regularisation leads to worse reconstructions but often better disentanglement [4, 7].

$$\mathcal{L}_{\text{rec}}(x, \hat{x}) = \mathbb{E}_{q_{\phi}(z|x)} [\log p_{\theta}(x|z)]; \quad \mathcal{L}_{\text{reg}}(x) = -D_{\text{KL}}(q_{\phi}(z|x) \parallel p_{\theta}(z)); \quad \mathcal{L}_{\text{VAE}}(x, \hat{x}) = \mathcal{L}_{\text{rec}}(x, \hat{x}) + \mathcal{L}_{\text{reg}}(x) \quad (2)$$

3 A Motivating Gridworld Environment

Before investigating representation learning in RL, we first consider the simpler unsupervised learning setting, in which VAEs and their variants are primarily developed through benchmarks over synthetic data. The 3D Shapes dataset [8] in Figure 1a contains observations of shapes fixed in the centre of the image with progressively changing attributes or factors such as size and colour. If, as humans, we are given unordered observations from a traversal along the size factor of 3D Shapes, it would be easy to order these observations using a perceived increase or decrease in the size of the shape. We might even say that the shapes in the images overlap by different amounts. We may consider shapes that are closer in size to possess more overlap, and therefore also consider them to be closer together in terms of distance. In contrast, consider a simple gridworld environment (Figure 1b) where an agent moves to adjacent cells in any of the four cardinal directions. The agent is a square of size 8×8 pixels and each action translates it 8 pixels in a given direction. Note that in this case, there is no overlap between the agent in any two images because of the distance it moves.

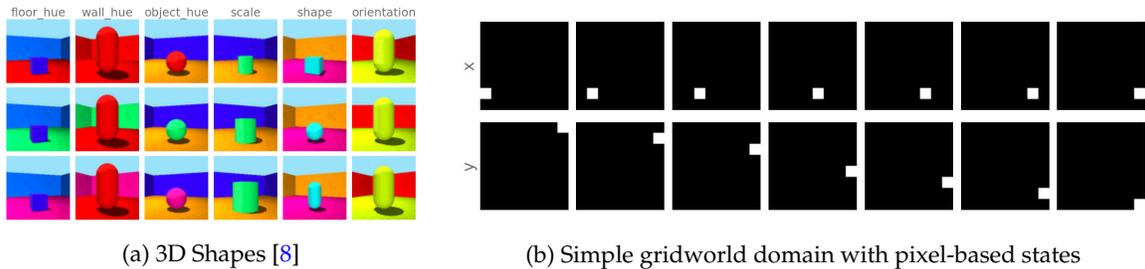


Figure 1: (a) A common synthetic dataset—images are generated by smoothly varying ground-truth factors such as floor hue and object shape. (b) Gridworld environment whose underlying factors are the agent’s xy -position.

It is therefore natural to ask whether a lack of overlap is problematic. To answer this question, let x be an observation and y be the underlying factors that generated it. We define the *ground-truth distance* between pairs of observations $\{x^{(a)}, x^{(b)}\}$ as the Manhattan distance between the observations’ underlying factors: $d_{\text{gt}}(x^{(a)}, x^{(b)}) = \|y^{(a)} - y^{(b)}\|_1$.² Similarly, we define the *perceived distance* as the difference between the two observations in pixel space as measured by the model’s reconstruction loss function (usually MSE): $d_{\text{vis}}(x^{(a)}, x^{(b)}) = \mathcal{L}_{\text{rec}}(x^{(a)}, x^{(b)})$.³

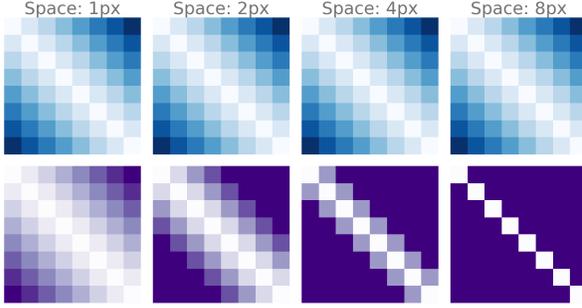
To investigate the above question, we modify the environment so that the agent moves in smaller increments. For each step size s , we generate all possible image states and store them in a buffer.⁴ As s decreases, the probability that the agent position overlaps in any two randomly sampled images increases. We next visualise ground-truth and perceived distances for each step size setting, with the results given by Figure 2a. The results show a clear relationship between the two distance measures—at very small step sizes (with high probability of overlap) there is almost perfect correspondence.

We further probe this relationship by using the data buffers as an unsupervised dataset. We make the problem harder by converting it to a multiagent environment which increases the number of states—three objects are now each described by their xy -positions. We train a β -VAE [7] and the state-of-the-art weakly-supervised Ada-GVAE [9]. The β -VAE scales the VAE regularisation term with a coefficient $\beta > 0$, while the Ada-GVAE encourages axis alignment and shared latent variables between pairs of observations. This is achieved by averaging together latent distributions between observation pairs that are estimated to remain unchanged when the KL divergence is below some threshold. To evaluate if the

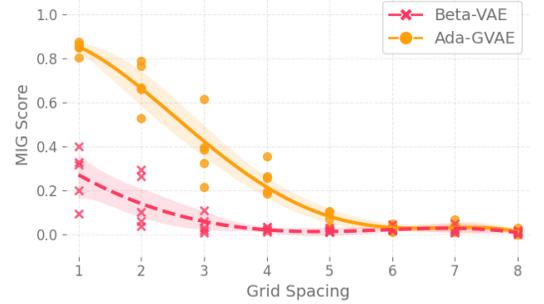
²The notation $\mathbf{a}^{(i)}$ is alternative notation for either a named variable or the i^{th} element of the ordered set \mathcal{A} .

³We use the term increased *perceived overlap* as a synonym for lower *perceived distance* between neighbouring states.

⁴We modify the size of the environment to ensure the size of the buffer remains constant regardless of the step size.



(a) Top Row: Manhattan ground-truth distance matrices over factor traversals. Bottom Row: Pixel-wise perceived distance matrices between observations over factor traversals. Left to Right: Step size increases from 1px to 8px. Smaller values result in more overlap in the data space, which leads to higher a probability of being able to find an ordering along a factor traversal.



(b) Regression plot of increasing domain spacing versus MIG score (higher is better). As the β -VAE (dashed line) and Ada-GVAE (solid line) are trained with decreasing (left to right) levels of overlap, their disentanglement performance worsens.

Figure 2: Varying spacing of valid locations in gridworld domain affects overlap as perceived by an VAE.

representations are disentangled, we use the Mutual Information Gap (MIG) [10]. The results in Figure 2b indicate that as the spacing decreases and overlap is introduced, the disentanglement performance improves as perceived pixel-wise distances better correspond to latent and ground-truth distances.

4 Triplet Loss

VAEs, therefore, appear to learn latent distances over the data that correspond to their reconstruction loss. However, if these distances are not useful, we may be able to instead guide the learning process by introducing *metric* or *similarity* learning. A common approach is triplet loss, which makes use of three observations $\mathcal{X}_{\text{triple}} = (\mathbf{x}^{(a)}, \mathbf{x}^{(p)}, \mathbf{x}^{(n)})$. These are sampled using supervision such that the *anchor* $\mathbf{x}^{(a)}$ is considered closer to *positive* $\mathbf{x}^{(p)}$ than it is to the *negative* $\mathbf{x}^{(n)}$. The intuition is that corresponding representations $\mathbf{z}^{(a)}, \mathbf{z}^{(p)}, \mathbf{z}^{(n)}$ output by a network should satisfy the original anchor-positive and anchor-negative distance constraints: $d(\mathbf{z}^{(a)}, \mathbf{z}^{(p)}) < d(\mathbf{z}^{(a)}, \mathbf{z}^{(n)})$. We construct the β -TVAE as three parallel versions of the same β -VAE augmented by the soft-margin formulation of triplet loss, where $\alpha > 0$ is the scaling factor:

$$\mathcal{L}_{\beta\text{TVAE}} = \underbrace{\alpha \ln \left(1 + \exp \left(d(\mathbf{z}^{(a)}, \mathbf{z}^{(p)}) - d(\mathbf{z}^{(a)}, \mathbf{z}^{(n)}) \right) \right)}_{\text{Triplet loss}} + \frac{1}{3} \sum_{\mathbf{x} \in \mathcal{X}_{\text{triple}}} \mathcal{L}_{\beta\text{VAE}}(\mathbf{x}) \quad (3)$$

4.1 Adaptive Triplet Loss

Standard triplet loss provides no direct pressure to learn factored representations; for example the x and y factors may be encoded with some arbitrary rotation in the latent space. To fix this, we adapt the Ada-GVAE[9] to construct Ada-Triplet and the Ada-TVAE. These adaptive methods encourage partially shared representations such that differences between observations are encoded in subsets of latent units. Shared latent units are estimated as those whose distances $\delta_i = |z_i^{(a)} - z_i^{(n)}|$ are less than half way between the maximum and minimum: $\delta_i < \frac{1}{2} (\min_i \delta_i + \max_i \delta_i)$. To encourage factored latents, we elementwise multiply \odot arguments of the anchor-negative triplet term by the weight vector $\mathbf{w} = (\omega_1, \dots, \omega_D)$. Shared latents are weighted less $\omega_i \in (0, 1)$ and non-shared units $\omega_i = 1$ remain unchanged:

$$\mathcal{L}_{\text{Ada-Triplet}} = \ln \left(1 + \exp \left(d(\mathbf{z}^{(a)}, \mathbf{z}^{(p)}) - d(\mathbf{w} \odot \mathbf{z}^{(a)}, \mathbf{w} \odot \mathbf{z}^{(n)}) \right) \right) \quad (4)$$

4.2 Using the Replay Buffer for Metric Learning and Feature Extraction

Typically, triplet loss is used in the supervised learning case, where the anchor, positive and negative samples are known. However, no such signal exists in the RL setting. To overcome this, we leverage the sequential nature of the problem and assume that *states closer together in time are also on average closer in terms of their ground-truth factors*.

Given a replay buffer $\mathcal{B} = \{s_0, a_0, r_0, s_1, a_1, r_1, \dots\}$, we use this assumption to sample some anchor state s_a , a positive s_p and a negative s_n such that $n > p > a$. To test the effectiveness of this assumption, we execute a uniformly random policy to populate a replay buffer. We train a β -VAE and Ada-GVAE as baselines, which we compare against the β -TVAE and Ada-TVAE from the previous sections. As a sanity check, we also train a fully supervised β -TVAE and Ada-TVAE on

the underlying ground-truth factors of the environment (the agent’s xy -position) so that the anchor-positive ℓ_1 distance is always less than the anchor-negative. The results in Figure 3a demonstrate that triplet-based models significantly outperform the baselines by learning better features.

Finally, we test the effect of using the encoders of these learnt models as feature extractors in an RL setting. We freeze the weights of the previously trained encoders, and apply deep Q-learning [2] over these features to solve the task of reaching the top left corner from the bottom right (rewards of -1 on all timesteps). Results in Figure 3b demonstrate that features from the adaptive triplet loss model are the most useful for the RL task; however, this is well approximated by sampling triplets using only the ordering of states from the replay buffer. Both triplet methods outperform the baselines; however, adaptive triplet encourages factored representations which are better features for the downstream RL task.

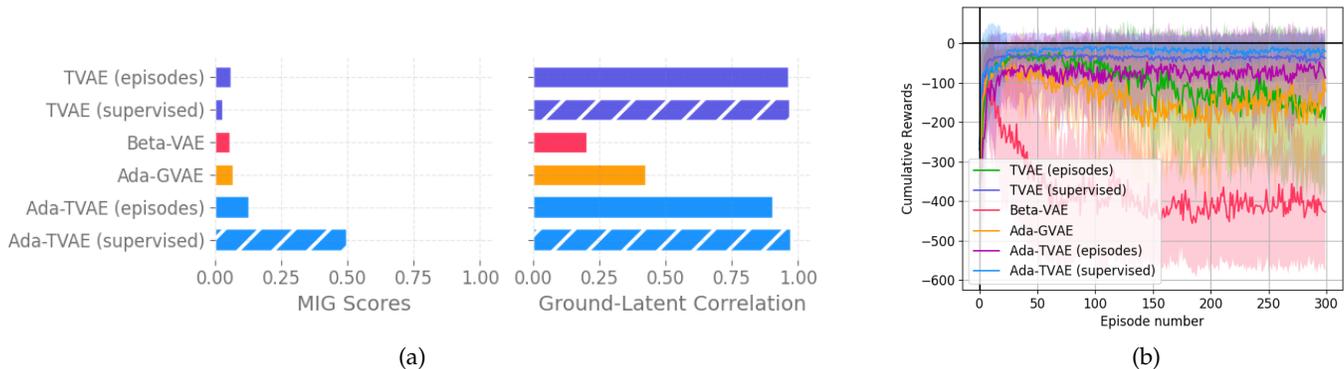


Figure 3: (a) Different VAE frameworks trained to extract features. The plot shows the average rank correlation between the ground-truth distances and corresponding distances between these learnt features. Striped bars are models trained using ground-truth distances, while solid bars use our approach of sampling sequentially from the replay buffer. (b) Downstream RL training rewards using previously learnt features. Mean and standard deviation are given over 30 runs.

5 Conclusion

We identified a shortcoming of VAE-based approaches in domains where there is little or no variation in perceived distances or overlap. This may be inherent in the environment, or arise inadvertently due to design decisions. For example, if frame skipping [2] or high-level skills are used, there may be no perceived overlap between successive states in the buffer. We overcome this problem by incorporating the inherent temporal information present in an episode through the simple use of metric learning. Furthermore, we improve the performance by enabling representation disentanglement by designing an adaptively weighted version of triplet loss. Leveraging the sequential nature of the replay buffer to perform metric learning may open up interesting new avenues for future feature learning approaches in RL.

References

- [1] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.
- [2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski et al., “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [3] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem, “Challenging common assumptions in the unsupervised learning of disentangled representations,” in *International Conference on Machine Learning*, 2019, pp. 4114–4124.
- [4] C. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, “Understanding disentangling in β -VAE,” in *Workshop on Learning Disentangled Representations at the 31st Conference on Neural Information Processing Systems*, 2017.
- [5] D. Kingma and M. Welling, “Auto-encoding variational bayes,” in *International Conference on Learning Representations*, 2014.
- [6] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [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,” 2016.
- [8] C. Burgess and H. Kim, “3D shapes dataset,” <https://github.com/deepmind/3dshapes-dataset/>, 2018.
- [9] F. Locatello, B. Poole, G. Rätsch, B. Schölkopf, O. Bachem, and M. Tschannen, “Weakly-supervised disentanglement without compromises,” in *International Conference on Machine Learning*, 2020, pp. 6348–6359.
- [10] R. Chen, X. Li, R. Grosse, and D. Duvenaud, “Isolating sources of disentanglement in variational autoencoders,” in *Advances in Neural Information Processing Systems*, 2018, pp. 2615–2625.