

---

# Life-Long Disentangled Representation Learning with Cross-Domain Latent Homologies

---

**Alessandro Achille, Tom Eccles, Loic Matthey, Christopher P Burgess,  
Nick Watters, Alexander Lerchner, Irina Higgins**

UCLA, DeepMind

achille@cs.ucla.edu,

{eccles, lmatthey, cpburgess, nwatters, lerchner, irinah}@google.com

## Abstract

Intelligent behaviour in the real-world requires the ability to acquire new knowledge from an ongoing sequence of experiences while preserving and reusing past knowledge. We propose a novel algorithm for unsupervised representation learning from piece-wise stationary visual data: Variational Autoencoder with Shared Embeddings (VASE). Based on the Minimum Description Length principle, VASE automatically detects shifts in the data distribution and allocates spare representational capacity to new knowledge, while simultaneously protecting previously learnt representations from catastrophic forgetting. Our approach encourages the learnt representations to be disentangled, which imparts a number of desirable properties: VASE can deal sensibly with ambiguous inputs, it can enhance its own representations through imagination-based exploration, and most importantly, it exhibits semantically meaningful sharing of latents between different datasets. Compared to baselines with entangled representations, our approach is able to reason beyond surface-level statistics and perform semantically meaningful cross-domain inference.

## 1 Introduction

A critical feature of biological intelligence is its capacity for *life-long learning* [10] – the ability to acquire new knowledge from a sequence of experiences to solve progressively more tasks, while maintaining performance on previous ones. This, however, remains a serious challenge for current deep learning approaches. While current methods are able to outperform humans on many individual problems [51, 36, 20], these algorithms suffer from *catastrophic forgetting* [14, 33, 34, 42, 17]. Training on a new task or environment can be enough to degrade their performance from super-human to chance level [46]. Another critical aspect of life-long learning is the ability to sensibly reuse previously learnt representations in new domains (*positive transfer*). For example, knowing that strawberries and bananas are not edible when they are green could be useful when deciding whether to eat a green peach in the future. Finding semantic homologies between visually distinctive domains can remove the need to learn from scratch on every new environment and hence help with data efficiency – another major drawback of current deep learning approaches [16, 29].

But how can an algorithm maximise the informativeness of the representation it learns on one domain for positive transfer on other domains without knowing *a priori* what experiences are to come? One approach might be to capture the important structure of the current environment in a maximally compact way (to preserve capacity for future learning). Such learning is likely to result in positive transfer if future training domains share some structural similarity with the old ones. This is a reasonable expectation to have for most natural (non-adversarial) tasks and environments, since they tend to adhere to the structure of the real world (e.g. relate to objects and their properties) governed by the consistent rules of chemistry or physics. A similar motivation underlies the Minimum Description Length (MDL) principle [44] and disentangled representation learning [8].

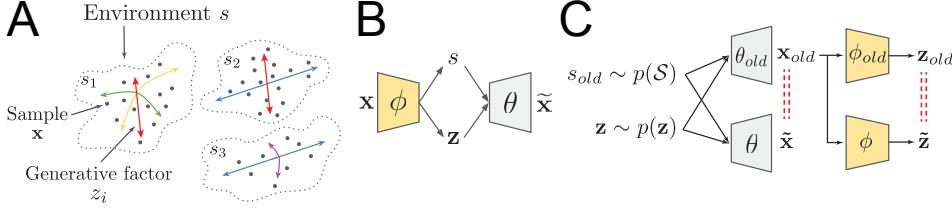


Figure 1: **A:** Schematic representation of the life-long learning data distribution. Each dataset/environment corresponds to a cluster  $s$ . Data samples  $x$  constituting each cluster can be described by a local set of coordinates (data generative factors  $z_n$ ). Different clusters may share some data generative factors. **B:** VASE model architecture **C:** ConContinSchematic of the “dreaming” feedback loop. We use a snapshot of the model with the old parameters ( $\phi_{old}$ ,  $\theta_{old}$ ) to generate an imaginary batch of data  $\mathbf{x}_{old}$  for a previously experienced dataset  $s_{old}$ . While learning in the current environment, we ensure that the representation is still consistent on the hallucinated “dream” data, and can reconstruct it (see red dashed lines).

Recent state of the art approaches to unsupervised disentangled representation learning [21, 9, 24, 28] use a modified Variational AutoEncoder (VAE) [26, 43] framework to learn a representation of the data generative factors. These approaches, however, only work on independent and identically distributed (IID) data from a single visual domain. This paper extends this line of work to life-long learning from piece-wise stationary data, exploiting this setting to learn shared representations across domains where applicable. The proposed Variational Autoencoder with Shared Embeddings (VASE, see fig. 1B) automatically detects shifts in the training data distribution and uses this information to allocate spare latent capacity to novel dataset-specific disentangled representations, while reusing previously acquired representations of latent dimensions where applicable. We use latent masking and a generative “dreaming” feedback loop (similar to [41, 49, 48, 5]) to avoid catastrophic forgetting. Our approach outperforms [41], the only other VAE based approach to life-long learning we are aware of. Furthermore, we demonstrate that the pressure to disentangle endows VASE with a number of useful properties: 1) dealing sensibly with ambiguous inputs; 2) learning richer representations through imagination-based exploration; 3) performing semantically meaningful cross-domain inference by ignoring irrelevant aspects of surface-level appearance.

## 2 Related work

The existing approaches to continual learning can be broadly separated into three categories: data-, architecture- or weights-based. The data-based approaches augment the training data on a new task with the data collected from the previous tasks, allowing for simultaneous multi-task learning on IID data [11, 45, 42, 33, 15]. The architecture-based approaches dynamically augment the network with new task-specific modules, which often share intermediate representations to encourage positive transfer [46, 39, 47]. Both of these types of approaches, however, are inefficient in terms of the memory requirements once the number of tasks becomes large. The weights-based approaches do not require data or model augmentation. Instead, they prevent catastrophic forgetting by slowing down learning in the weights that are deemed to be important for the previously learnt tasks [27, 53, 38]. This is a promising direction, however, its application is limited by the fact that it typically uses knowledge of the task presentation schedule to update the loss function after each switch in the data distribution.

Most of the continual learning literature, including all of the approaches discussed above, have been developed in task-based settings, where representations are learnt implicitly. While deep networks learn well in such settings [1, 50], this often comes at a cost of reduced positive transfer. This is because the implicitly learnt representations often overfit to the training task by discarding information that is irrelevant to the current task but may be required for solving future tasks [1–3, 50, 22]. The acquisition of useful representations of complex high-dimensional data without task-based overfitting is a core goal of unsupervised learning. Past work [2, 4, 21] has demonstrated the usefulness of information-theoretic methods in such settings. These approaches can broadly be seen as efficient implementations of the Minimum Description Length (MDL) principle for unsupervised learning [44, 18]. The representations learnt through such methods have been shown to help in transfer scenarios and with data efficiency for policy learning in the Reinforcement Learning (RL) context [22]. These approaches, however, do not immediately generalise to non-stationary data. Indeed, life-long unsupervised representation learning is relatively under-developed [49, 48, 38]. The majority of recent work in this direction has concentrated on implicit generative models [49, 48], or non-parametric approaches [35]. Since these

approaches do not possess an inference mechanism, they are unlikely to be useful for subsequent task or policy learning. Furthermore, none of the existing approaches explicitly investigate meaningful sharing of latent representations between environments.

### 3 Framework

#### 3.1 Problem formalisation

We assume that there is an a priori unknown set  $\mathcal{S} = \{s_1, s_2, \dots, s_K\}$  of  $K$  environments which, between them, share a set  $\mathcal{Z} = \{z_1, z_2, \dots, z_N\}$  of  $N$  independent data generative factors. We assume  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ . Since we aim to model piece-wise stationary data, it is reasonable to assume  $s \sim \text{Cat}(\pi_{1, \dots, K})$ , where  $\pi_k$  is the probability of observing environment  $s_k$ . Two environments may use the same generative factors but render them differently, or they may use a different subset of factors altogether. Given an environment  $s$ , and an environment-dependent subset  $\mathcal{Z}^s \subseteq \mathcal{Z}$  of the ground truth generative factors, it is possible to synthesise a dataset of images  $\mathbf{x}^s \sim p(\cdot | \mathbf{z}^s, s)$ . In order to keep track of which subset of the  $N$  data generative factors is used by each environment  $s$  to generate images  $\mathbf{x}^s$ , we introduce an environment-dependent mask  $\mathbf{a}^s$  with dimensionality  $|\mathbf{a}| = N$ , where  $a_n^s = 1$  if  $z_n \in \mathcal{Z}^s$  and zero otherwise. Hence, we assume  $\mathbf{a}^s \sim \text{Bern}(\omega_{1, \dots, N}^s)$ , where  $\omega_n^s$  is the probability that factor  $z_n$  is used in environment  $s$ . This leads to the following generative process (where “ $\odot$ ” is element-wise multiplication):

$$\begin{aligned} \mathbf{z} &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}), & s &\sim \text{Cat}(\pi_{1, \dots, K}), & \mathbf{a}^s &\sim \text{Bern}(\omega_{1, \dots, N}^s), \\ \mathbf{z}^s &= \mathbf{a}^s \odot \mathbf{z}, & \mathbf{x}^s &\sim p(\cdot | \mathbf{z}^s, s) \end{aligned} \quad (1)$$

Intuitively, we assume that the piece-wise stationary observed data  $\mathbf{x}$  can be split into *clusters* (environments  $s$ ) (note evidence for similar experience clustering from the animal literature [6]). Each cluster has a set of *standard coordinate axes* (a subset of the generative factors  $\mathbf{z}$  chosen by the latent mask  $\mathbf{a}^s$ ) that can be used to parametrise the data in that cluster (fig. 1A). Given a sequence  $\mathbf{x} = (\mathbf{x}^{s_1}, \mathbf{x}^{s_2}, \dots)$  of datasets generated according to the process in eq. (1), where  $s_k \sim p(s)$  is the  $k$ -th sample of the environment, the aim of life-long representation learning can be seen as estimating the full set of generative factors  $\mathcal{Z} \approx \bigcup_k q(\mathbf{z}^{s_k} | \mathbf{x}^{s_k})$  from the environment-specific subsets of  $\mathbf{z}$  inferred on each stationary data cluster  $\mathbf{x}^{s_k}$ . Henceforth, we will drop the subscript  $k$  for simplicity of notation.

#### 3.2 Inferring the data generative factors

Observations  $\mathbf{x}^s$  cannot contain information about the generative factors  $z_n$  that are not relevant for the environment  $s$ . Hence, we use the following form for representing the data generative factors:

$$q(\mathbf{z}^s | \mathbf{x}^s) = \mathbf{a}^s \odot \mathcal{N}(\mu(\mathbf{x}), \sigma(\mathbf{x})) + (1 - \mathbf{a}^s) \odot \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (2)$$

Note that  $\mu$  and  $\sigma$  in eq. (2) depend only on the data  $\mathbf{x}$  and not on the environment  $s$ . This is important to ensure that the semantic meaning of each latent dimension  $z_n$  remains consistent for different environments  $s$ . We model the representation  $q(\mathbf{z}^s | \mathbf{x}^s)$  of the data generative factors as a product of independent normal distributions to match the assumed prior  $p(\mathbf{z}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ .

In order to encourage the representation  $q(\mathbf{z}^s | \mathbf{x}^s)$  to be semantically meaningful, we encourage it to capture the generative factors of variation within the data  $\mathbf{x}^s$  by following the MDL principle. We aim to find a representation  $\mathbf{z}^s$  that minimises the reconstruction error of the input data  $\mathbf{x}^s$  conditioned on  $\mathbf{z}^s$  under a constraint on the quantity of information in  $\mathbf{z}^s$ . This leads to the following loss function:

$$\mathcal{L}_{\text{MDL}}(\phi, \theta) = \underbrace{\mathbb{E}_{\mathbf{z}^s \sim q_\phi(\cdot | \mathbf{x}^s)}[-\log p_\theta(\mathbf{x} | \mathbf{z}^s, s)]}_{\text{Reconstruction error}} + \gamma \underbrace{\mathbb{E}_{\mathbf{z}^s \sim q_\phi(\cdot | \mathbf{x}^s)}[\mathbb{K}\mathbb{L}(q_\phi(\mathbf{z}^s | \mathbf{x}^s) || p(\mathbf{z}))]}_{\text{Representation capacity}} - \underbrace{C}_{\text{Target}}^2 \quad (3)$$

The loss in eq. (3) is closely related to the  $\beta$ -VAE [21] objective  $\mathcal{L} = \mathbb{E}_{\mathbf{z} \sim q_\phi(\cdot | \mathbf{x})}[-\log p_\theta(\mathbf{x} | \mathbf{z})] + \beta \mathbb{K}\mathbb{L}(q_\phi(\mathbf{z} | \mathbf{x}) || p(\mathbf{z}))$ , which uses a Lagrangian to limit the latent bottleneck capacity, rather than an explicit target  $C$ . It was shown that optimising the  $\beta$ -VAE objective helps with learning a more semantically meaningful disentangled representation  $q(\mathbf{z} | \mathbf{x})$  of the data generative factors [21]. However, [9] showed that progressively increasing the target capacity  $C$  in eq. (3) throughout training further improves the disentanglement results reported in [21], while simultaneously producing sharper reconstructions. Progressive increase of the representational capacity also seems intuitively better suited to continual learning where new information is introduced in a sequential manner. Hence, VASE optimises the objective function in eq. (3) over a sequence of datasets  $\mathbf{x}^s$ . This, however, requires a way to infer  $s$  and  $\mathbf{a}^s$ , as discussed next.

### 3.3 Inferring the latent mask

Given a dataset  $\mathbf{x}^s$ , we want to infer which latent dimensions  $z_n$  were used in its generative process (see eq. (1)). This serves multiple purposes: 1) helps identify the environment  $s$  (see next section); 2) helps ignore latent factors  $z_n$  that encode useful information in some environment but are not used in the current environment  $s$ , in order to prevent retraining and subsequent catastrophic forgetting; and 3) promotes latent sharing between environments. Remember that eq. (3) indirectly optimises for  $\mathbb{E}_{\mathbf{x}^s}[q_\phi(\mathbf{z}^s|\mathbf{x}^s)] \approx p(\mathbf{z})$  after training on a dataset  $s$ . If a new dataset uses the same generative factors as  $\mathbf{x}^s$ , then the marginal behaviour of the corresponding latent dimensions  $z_n$  will not change. On the other hand, if a latent dimension encodes a data generative factor that is irrelevant to the new dataset, then it will start behaving atypically and stray away from the prior. We capture this intuition by defining the *atypicality* score  $\alpha_n$  for each latent dimension  $z_n$  on a batch of data  $\mathbf{x}_{\text{batch}}^s$ :

$$\alpha_n = \mathbb{KL}(\mathbb{E}_{\mathbf{x}_{\text{batch}}^s}[q_\phi(z_n^s|\mathbf{x}_{\text{batch}}^s)] \parallel p(z_n)). \quad (4)$$

The atypical components are unlikely to be relevant to the current environment, so we mask them out:

$$a_n^s = \begin{cases} 1, & \text{if } \alpha_n < \lambda \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where  $\lambda$  is a threshold hyperparameter (see appendices A.2 and A.3 for more details). Note that the uninformative latent dimensions  $z_n$  that have not yet learnt to represent any data generative factors, i.e.  $q_\phi(z_n|\mathbf{x}_n^s) = p(z_n)$ , are automatically unmasked in this setup. This allows them to be available as spare latent capacity to learn new generative factors when exposed to a new dataset. Fig. 2 shows the sharp changes in  $\alpha_n$  at dataset boundaries during training.

### 3.4 Inferring the environment

Given the generative process introduced in eq. (1), it may be tempting to treat the environment  $s$  as a discrete latent variable and learn it through amortised variational inference. However, we found that in the continual learning scenario this is not a viable strategy. Parametric learning is slow, yet we have to infer each new data cluster  $s$  extremely fast to avoid catastrophic forgetting. Hence, we opt for a fast non-parametric meta-algorithm motivated by the following intuition. Having already experienced  $r$  datasets during life-long learning, there are two choices when it comes to inferring the current one  $s$ : it is either a new dataset  $s_{r+1}$ , or it is one of the  $r$  datasets encountered in the past. Intuitively, one way to check for the former is to see whether the current data  $\mathbf{x}^s$  seems likely under any of the previously seen environments. This condition on its own is not sufficient though. It is possible that environment  $s$  uses a subset of the generative factors used by another environment  $\mathcal{Z}^s \subseteq \mathcal{Z}^t$ , in which case environment  $t$  will explain the data  $\mathbf{x}^s$  well, yet it will be an incorrect inference. Hence, we have to ensure that the subset of the relevant generative factors  $\mathbf{z}^s$  inferred for the current data  $\mathbf{x}^s$  according to section 3.3 matches that of the candidate past dataset  $t$ . Given a batch  $\mathbf{x}_{\text{batch}}^s$ , we infer the environment  $s$  according to:

$$s = \begin{cases} \hat{s}, & \text{if } \mathbb{E}_{\mathbf{z}^{\hat{s}}}[p_\theta(\mathbf{x}_{\text{batch}}^s|\mathbf{z}^{\hat{s}}, \hat{s})] \leq \kappa L_{\hat{s}} \wedge \mathbf{a}^s = \mathbf{a}^{\hat{s}} \\ s_{r+1}, & \text{otherwise} \end{cases} \quad (6)$$

where  $\hat{s} = \operatorname{argmax}_s q(s|\mathbf{x}_{\text{batch}}^s)$  is the output of an auxiliary classifier trained to infer the most likely previously experienced environment  $\hat{s}$  given the current batch  $\mathbf{x}_{\text{batch}}^s$ ,  $L_{\hat{s}}$  is the average reconstruction error observed for the environment  $\hat{s}$  when it was last experienced, and  $\kappa$  is a threshold hyperparameter (see appendix A.2 for details).

### 3.5 Preventing catastrophic forgetting

So far we have discussed how VASE integrates knowledge from the current environment into its representation  $q_\phi(\mathbf{z}|\mathbf{x})$ , but we haven't yet discussed how we ensure that past knowledge is not forgotten in the process. Most standard approaches to preventing catastrophic forgetting discussed in section 2 are either not applicable to a variational context, or do not scale well due to memory requirements. However, thanks to learning a generative model of the observed environments, we can prevent catastrophic forgetting by periodically *hallucinating* (i.e. generating samples) from past environments using a snapshot of VASE, and making sure that the current version of VASE is still able to model these samples. A similar “dreaming” feedback loop was used in [41, 49, 48, 5].

More formally, we follow the generative process in eq. (1) to create a batch of samples  $\mathbf{x}_{\text{old}} \sim q_{\theta_{\text{old}}}(\cdot | \mathbf{z}, s_{\text{old}})$  using a snapshot of VASE with parameters  $(\phi_{\text{old}}, \theta_{\text{old}})$  (see fig. 1C). We then update the current version of VASE according to the following (replacing  $_{\text{old}}$  with  $'$  for brevity):

$$\mathcal{L}_{\text{past}}(\phi, \theta) = \mathbb{E}_{\mathbf{z}, s', \mathbf{x}'} \left[ \underbrace{D[q_\phi(\mathbf{z} | \mathbf{x}'), q_{\phi'}(\mathbf{z}' | \mathbf{x}')]}_{\text{Encoder proximity}} + \underbrace{D[q_\theta(\mathbf{x} | \mathbf{z}, s'), q_{\theta'}(\mathbf{x}' | \mathbf{z}, s')]}_{\text{Decoder proximity}} \right], \quad (7)$$

where  $D$  is a distance between two distributions (we use the Wasserstein distance for the encoder and KL divergence for the decoder). The snapshot parameters get synced to the current trainable parameters  $\phi_{\text{old}} \leftarrow \phi, \theta_{\text{old}} \leftarrow \theta$  every  $\tau$  training steps, where  $\tau$  is a hyperparameter. The expectation over simulators  $s_{\text{old}}$  and latents  $\mathbf{z}$  in eq. (7) is done using Monte Carlo sampling (see appendix A.2 for details).

### 3.6 Model summary

To summarise, we train our model using a meta-algorithm with both parametric and non-parametric components. The latter is needed to quickly associate new experiences to an appropriate cluster, so that learning can happen inside the current experience cluster, without disrupting unrelated clusters. We initialise the latent representation  $\mathbf{z}$  to have at least as many dimensions as the total number of the data generative factors  $|\mathbf{z}| \geq |\mathcal{Z}| = N$ , and the softmax layer of the auxiliary environment classifier to be at least as large as the number of datasets  $|\mathcal{S}| = K$ . As we observe the sequence of training data, we detect changes in the environment and dynamically update the internal estimate of  $r \leq K$  datasets experienced so far according to eq. (6). We then train VASE by minimising the following objective function:

$$\begin{aligned} \mathcal{L}(\phi, \theta) = & \underbrace{\mathbb{E}_{\mathbf{z}^s \sim q_\phi(\cdot | \mathbf{x}^s)} [-\log p_\theta(\mathbf{x} | \mathbf{z}^s, s)] + \gamma \underbrace{\| \mathbb{KLL}(q_\phi(\mathbf{z}^s | \mathbf{x}^s) || p(\mathbf{z})) - C \|^2}_{\text{MDL on current data}} +} \\ & + \underbrace{\mathbb{E}_{\mathbf{z}, s', \mathbf{x}'} \left[ D[q_\phi(\mathbf{z} | \mathbf{x}'), q_{\phi'}(\mathbf{z}' | \mathbf{x}')] + D[q_\theta(\mathbf{x} | \mathbf{z}, s'), q_{\theta'}(\mathbf{x}' | \mathbf{z}, s')] \right]}_{\text{"Dreaming" feedback on past data}}. \end{aligned} \quad (8)$$

## 4 Experiments

**Continual learning with disentangled shared latents** First, we qualitatively assess whether VASE is able to learn good representations in a continual learning setup. We use a sequence of three datasets: (1) a moving version of Fashion-MNIST [52] (shortened to moving Fashion), (2) MNIST [30], and (3) a moving version of MNIST (moving MNIST). During training we expect VASE to detect shifts in the data distribution and dynamically create new experience clusters  $s$ , learn a disentangled representation of each environment without forgetting past environments, and share disentangled factors between environments in a semantically meaningful way. Fig. 2 (top) compares the performance of VASE to that of Controlled Capacity Increase-VAE (CCI-VAE) [9], a model for disentangled representation learning with the same architecture as VASE but without the modifications introduced in this paper to allow for continual learning. It can be seen that unlike VASE, CCI-VAE forgot moving Fashion at the end of the training sequence. Both models were able to disentangle position from object identity, however, only VASE was able to meaningfully share latents between the different datasets - the two positional latents are active for two moving datasets but not for the static MNIST. VASE also has moving Fashion- and MNIST-specific latents, while CCI-VAE shares all latents between all datasets. VASE use only 8/24 latent dimensions at the end of training. The rest remained as spare capacity for learning on future datasets.

**Learning representations for tasks** We train object identity classifiers (one each for moving Fashion and MNIST) and an object position regressor on top of the latent representation  $\mathbf{z} \sim q_\phi(\mathbf{z} | \mathbf{x})$  at regular intervals throughout the continual learning sequence. Good accuracy on these measures would indicate that at the point of measurement, the latent representation  $\mathbf{z}$  contained dataset relevant information, and hence could be useful, e.g. for subsequent policy learning in RL agents. Figure 2 (bottom) shows that both VASE and CCI-VAE learn progressively more informative latent representations when exposed to each dataset  $s$ , as evidenced by the increasing classification accuracy and decreasing mean squared error (MSE) measures within each stage of training. However, with CCI-VAE, the accuracy and MSE measures degrade sharply once a domain shift occurs. This is not the case for VASE, which retains a relatively stable representation.

## VASE

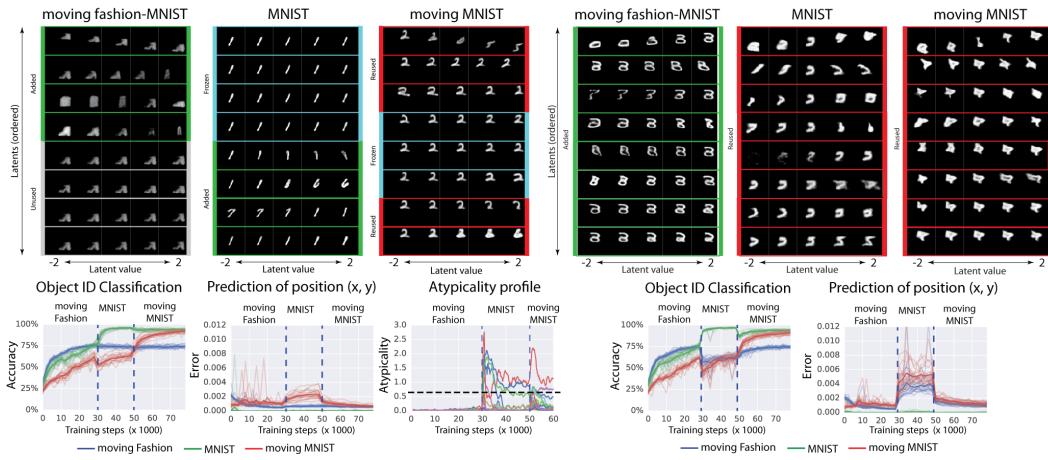


Figure 2: We compare VASE to a CCI-VAE baseline. Both are trained on a sequence of three datasets: moving fashion MNIST (moving Fashion) → MNIST → moving MNIST. **Top:** latent traversals at the end of training seeded with samples from the three datasets. The value of each latent  $z_n$  is traversed between -2 and 2 one at a time, and the corresponding reconstructions are shown. Rows correspond to latent dimensions  $z_n$ , columns correspond to the traversal values. Latent use progression throughout training is demonstrated in colour. **Bottom:** performance of MNIST and Fashion object classifiers and a position regressor trained on the latent space  $\mathbf{z}$  throughout training. Note the relative stability of the curves for VASE compared to the baseline. The atypicality profile shows the values of  $\alpha_n$  through training (different colours indicate different latent dimensions), with the threshold  $\lambda$  indicated by the dashed black line.

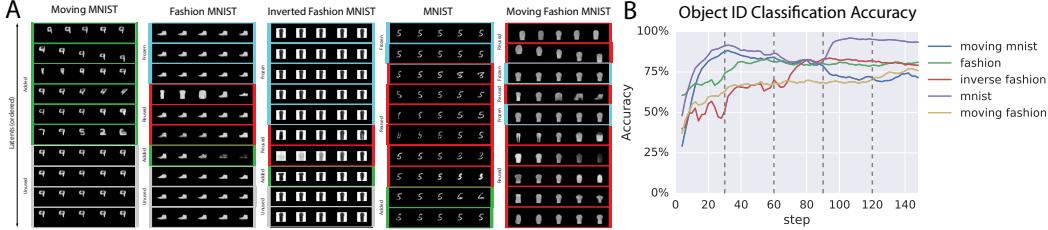


Figure 3: Latent traversals (A) and classification accuracy (B) (both as in fig. 2) for VASE trained on a sequence of moving MNIST → Fashion → inverse Fashion → MNIST → moving Fashion. See fig. 7 for larger traversals.

**Ablation study** Here we perform a full ablation study to test the importance of the proposed components for unsupervised life-long representation learning: 1) regularisation towards disentangled representations (section 3.2), 2) latent masking (section 3.3 - A), 3) environment clustering (section 3.4 - S), and 4) “dreaming” feedback loop (section 3.5 - D). We use the constraint capacity loss in eq. (3) for the disentangled experiments, and the standard VAE loss [26, 43] for the entangled experiments [21]. For each condition we report the average change in the classification metrics reported above, and the average maximum values achieved (see appendix A.5 for details). Table 1 shows that the unablated VASE (SDA) has the best performance. Note that the entangled baselines perform worse than the disentangled equivalents, and that the capacity constraint of the CCI-VAE framework does not significantly affect the maximal classification accuracy compared to the VAE. It is also worth noting that VASE outperforms the entangled SD condition, which is similar to the only other baseline VAE-base approach to continual learning that we are aware of [41]. We have also trained VASE on longer sequences of datasets (moving MNIST → Fashion → inverse Fashion → MNIST → moving Fashion) and found similar levels of performance (see fig. 3).

**Dealing with ambiguity** Natural stimuli are often ambiguous and may be interpreted differently based on contextual clues. Examples of such processes are common, e.g. visual illusions like the Necker cube [37], and may be driven by the functional organisation and the heavy top-down influences within the ventral visual stream of the brain [19, 40]. To evaluate the ability of VASE to deal with ambiguous inputs based on the context, we train it on a CelebA [32] → inverse Fashion sequence, and

ABLATION	DISENTANGLED				ENTANGLED			
	OBJECT ID ACCURACY		POSITION MSE		OBJECT ID ACCURACY		POSITION MSE	
	MAX (%)	CHANGE (%)	MIN (*1E-4)	CHANGE (*1E-4)	MAX (%)	CHANGE (%)	MIN (*1E-4)	CHANGE (*1E-4)
-	88.6 ( $\pm 0.4$ )	-15.2 ( $\pm 2.8$ )	3.5 ( $\pm 0.05$ )	24.8 ( $\pm 13.5$ )	91.8 ( $\pm 0.4$ )	-12.1 ( $\pm 0.8$ )	4.2 ( $\pm 0.7$ )	10.5 ( $\pm 2.6$ )
S	88.9 ( $\pm 0.5$ )	-13.9 ( $\pm 1.9$ )	3.4 ( $\pm 0.05$ )	22.5 ( $\pm 12.2$ )	91.7 ( $\pm 0.4$ )	-12.2 ( $\pm 0.03$ )	4.5 ( $\pm 0.8$ )	10.9 ( $\pm 3.1$ )
D	88.6 ( $\pm 0.3$ )	-14.4 ( $\pm 1.9$ )	3.3 ( $\pm 0.04$ )	21.4 ( $\pm 4.9$ )	91.8 ( $\pm 0.4$ )	-12.4 ( $\pm 0.7$ )	4.3 ( $\pm 0.7$ )	11.7 ( $\pm 3.2$ )
A	86.7 ( $\pm 1.9$ )	-24.5 ( $\pm 1.0$ )	3.3 ( $\pm 0.04$ )	67.6 ( $\pm 107.0$ )	88.6 ( $\pm 0.3$ )	-19.7 ( $\pm 0.5$ )	4.5 ( $\pm 0.7$ )	47.1 ( $\pm 26.2$ )
SA	87.1 ( $\pm 1.8$ )	-28.1 ( $\pm 0.08$ )	3.3 ( $\pm 0.04$ )	78.9 ( $\pm 109.0$ )	89.9 ( $\pm 1.3$ )	-18.3 ( $\pm 0.4$ )	4.8 ( $\pm 0.7$ )	41.8 ( $\pm 20.6$ )
DA	86.3 ( $\pm 2.5$ )	-25.2 ( $\pm 0.5$ )	3.3 ( $\pm 0.04$ )	72.2 ( $\pm 90.0$ )	88.8 ( $\pm 0.3$ )	-19.4 ( $\pm 0.4$ )	4.6 ( $\pm 0.7$ )	40.2 ( $\pm 19.2$ )
SD	88.3 ( $\pm 0.3$ )	-12.9 ( $\pm 1.9$ )	3.4 ( $\pm 0.05$ )	20.0 ( $\pm 3.5$ )	91.4 ( $\pm 0.3$ )	-11.7 ( $\pm 0.6$ )	4.3 ( $\pm 0.5$ )	11.6 ( $\pm 1.9$ )
<b>VASE (SDA)</b>	<b>88.6 (<math>\pm 0.4</math>)</b>	<b>-5.4 (<math>\pm 0.3</math>)</b>	<b>3.2 (<math>\pm 0.03</math>)</b>	<b>3.0 (<math>\pm 0.2</math>)</b>	<b>91.5 (<math>\pm 0.1</math>)</b>	<b>-6.5 (<math>\pm 0.7</math>)</b>	<b>4.2 (<math>\pm 0.4</math>)</b>	<b>3.9 (<math>\pm 1.1</math>)</b>

Table 1: Average change in classification accuracy/MSE and maximum/minimum average accuracy/MSE when training an object/position classifier/regressor on top of the learnt representation on the moving Fashion → MNIST → moving MNIST sequence. We do a full ablation study of VASE, where D - dreaming feedback loop, S - cluster inference  $q(s|x^s)$ , and A - atypicality based latent mask  $a^s$  inference. We compare two versions of our model - one that is encouraged to learn a disentangled representation through the capacity increase regularisation in eq. (3), and an entangled VAE baseline ( $\beta = 1$ ). The unablated disentangled version of VASE (SDA) has the best performance.

test it using ambiguous linear interpolations between samples from the two datasets (fig. 4A, first row). To measure the effects of ambiguity, we varied the interpolation weights between the two datasets. To measure the effects of context, we presented the ambiguous samples in a batch with real samples from one of the training datasets, varying the relative proportions of the two. Figure 4A (bottom) shows the inferred probability of interpreting the ambiguous samples as CelebA  $q_\phi(s = \text{celebA}|\mathbf{x})$ . VASE shows a sharp boundary between interpreting input samples as Fashion or CelebA despite smooth changes in input ambiguity. Such *categorical perception* is also characteristic of biological intelligence [12, 13, 31]. The decision boundary for categorical perception is affected by the context in which the ambiguous samples are presented. VASE also represents its uncertainty about the ambiguous inputs by increasing the inferred variance of the relevant latent dimensions (fig. 4A, second row).

**Semantic transfer** Here we test whether VASE can learn more sophisticated cross-domain latent homologies than the positional latents on the moving MNIST and Fashion datasets described above. Hence, we trained VASE on a sequence of two visually challenging DMLab-30<sup>1</sup> [7] datasets: the Exploit Deferred Effects (EDE) environment and a randomized version of the Natural Labyrinth (NatLab) environment (Varying Map Randomized). While being visually very distinct (one being indoors and the other outdoors), the two datasets share many data generative factors that have to do with the 3D geometry of the world (e.g. horizon, walls/terrain, objects/cacti) and the agent’s movements (first person optic flow). Hence, the two domains share many semantically related factors  $\mathbf{z}$ , but these are rendered into very different visuals  $\mathbf{x}$ . We compared cross-domain reconstructions of VASE and an equivalent entangled VAE ( $\beta = 1$ ) baseline. The reconstructions were produced by first inferring a latent representation based on a batch from one domain, e.g.  $\mathbf{z}^{\text{NatLab}} = q_\phi(\cdot | \mathbf{x}^{\text{NatLab}})$ , and then reconstructing them conditioned on the other domain  $\mathbf{x}^{\text{Rec}} = q_\theta(\cdot | \mathbf{z}^{\text{NatLab}}, s^{\text{EDE}})$ . Fig. 4 shows that VASE discovered the latent homologies between the two domains, while the entangled baseline failed to do so. VASE learnt the semantic equivalence between the cacti in NatLab and the red objects in EDE, the brown fog corresponding to the edge of the NatLab world and the walls in EDE (top leftmost reconstruction), and the horizon lines in both domains. The entangled baseline, on the other hand, seemed to rely on the surface-level pixel statistics and hence struggled to produce meaningful cross-domain reconstructions, attempting to match the texture rather than the semantics of the other domain. See appendix A.6 for additional cross-domain reconstructions, including on the sequence of five datasets mentioned earlier.

**Imagination-driven exploration** Once we learn the concept of moving objects in one environment, it is reasonable to imagine that a novel object encountered in a different environment can also be moved. Given the ability to act, we may try to move the object to realise our hypothesis. We can use such imagination-driven exploration to augment our experiences in an environment and let us learn a richer representation. Notice however, that such imagination requires a compositional representation that allows for novel yet sensible recombinations of previously learnt semantic factors. We now investigate whether VASE can use such imagination-driven exploration to learn better representations using a sequence of three datasets: moving Fashion → MNIST → moving MNIST. During the first moving Fashion stage, VASE learns the concepts of position and Fashion sprites. It also learns how to move the sprites to reach the imagined states  $z^*$  by training an auxiliary policy (see appendix A.3.1 for details).

<sup>1</sup>[https://github.com/deepmind/lab/tree/master/game\\_scripts/levels/contributed/dmlab30#dmlab-30](https://github.com/deepmind/lab/tree/master/game_scripts/levels/contributed/dmlab30#dmlab-30)

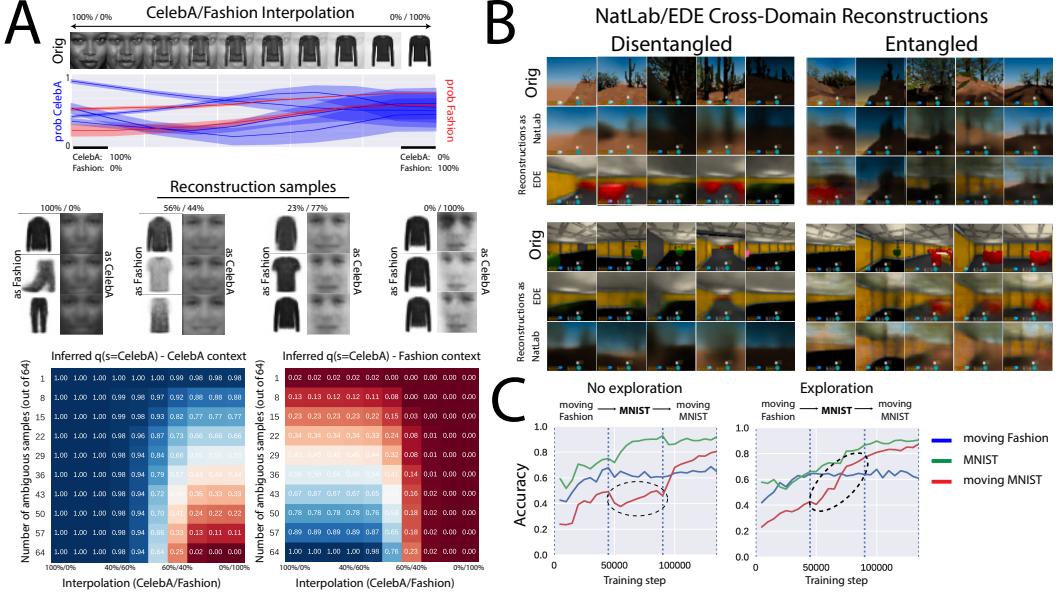


Figure 4: **A** Top: Ambiguous input examples created by using different interpolation weights between samples from CelebA and Fashion, and corresponding inferred parameters  $\mu$  (y axis) and  $\sigma$  (light colour range) of  $q_\phi(\mathbf{z}|\mathbf{x})$ ; red corresponds to Fashion-specific latents, blue to CelebA-specific latents. Middle: Reconstruction samples  $p_\theta(\mathbf{x}^s|\mathbf{z}^s, s)$  for different levels of ambiguity conditioned on either dataset. Bottom: Inferred  $q_\psi(s = \text{CelebA}$  given different levels of input ambiguity (x axis) and different number of ambiguous vs real data samples (y axis) for the two datasets. VASE deals well with ambiguity, shows context-dependent categorical perception and uncertainty within its inferred representation parameters. **B** Cross-domain reconstructions on NatLab (outdoors) or EDE (indoors) DM Lab levels. The disentangled VASE finds semantic homologies between the two datasets (e.g. cacti  $\rightarrow$  red objects). The entangled VASE only maps lower level statistics. **C** Imagination-based exploration allows VASE to imagine the possibility of moving MNIST digits during static MNIST training by using position latents acquired on moving Fashion. This helps it learn a moving MNIST classifier during static MNIST training without ever seeing real translations of MNIST digits.

It can then use this policy to do an imagination-based augmentation of the input data on MNIST by imagining MNIST digits in different positions and transforming the static sprites correspondingly using the learnt policy. Hence, VASE can imagine the existence of moving MNIST before actually experiencing it. Indeed, fig. 4C shows that when we train a moving MNIST classifier during the static MNIST training stage, the classifier is able to achieve good accuracy in the *imagination-driven exploration* condition, highlighting the benefits of imagination-driven data augmentation.

## 5 Conclusions

We have introduced VASE, a novel approach to life-long unsupervised representation learning that builds on recent work on disentangled factor learning [21, 9] by introducing several new key components. Unlike other approaches to continual learning, our algorithm does not require us to maintain a replay buffer of past datasets, or to change the loss function after each dataset switch. In fact, it does not require any a priori knowledge of the dataset presentation sequence, since these changes in data distribution are automatically inferred. We have demonstrated that VASE can learn a disentangled representation of a sequence of datasets. It does so without experiencing catastrophic forgetting and by dynamically allocating spare capacity to represent new information. It resolves ambiguity in a manner that is analogous to the categorical perception characteristic of biological intelligence. Most importantly, VASE allows for semantically meaningful sharing of latents between different datasets, which enables it to perform cross-domain inference and imagination-driven exploration. Taken together, these properties make VASE a promising algorithm for learning representations that are conducive to subsequent robust and data-efficient RL policy learning.

## References

- [1] A. Achille and S. Soatto. Emergence of Invariance and Disentangling in Deep Representations. *Proceedings of the ICML Workshop on Principled Approaches to Deep Learning*, 2017.
- [2] A. Achille and S. Soatto. Information dropout: Learning optimal representations through noisy computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2018.
- [3] A. Achille and S. Soatto. A separation principle for control in the age of deep learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):null, 2018.
- [4] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.
- [5] B. Ans and S. Rousset. Avoiding catastrophic forgetting by coupling two reverberating neural networks. *Comptes Rendus de l'Académie des Sciences - Series III - Sciences de la Vie*, 320(12):989–997, 1997.
- [6] A. Auchter, L. K. Cormack, Y. Niv, F. Gonzalez-Lima, and M. H. Monfils. Reconsolidation-extinction interactions in fear memory attenuation: the role of inter-trial interval variability. *Frontiers in behavioral neuroscience*, 11:2, 2017.
- [7] C. Beattie, J. Z. Leibo, D. Teplyashin, T. Ward, M. Wainwright, H. Küttler, A. Lefrancq, S. Green, V. Valdés, A. Sadik, J. Schrittwieser, K. Anderson, S. York, M. Cant, A. Cain, A. Bolton, S. Gaffney, H. King, D. Hassabis, S. Legg, and S. Petersen. Deepmind lab. *arXiv preprint arXiv:1612.03801*, 2016.
- [8] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.
- [9] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. Understanding disentangling in  $\beta$ -VAE. *NIPS Workshop of Learning Disentangled Features*, 2017.
- [10] J. Cichon and W.-B. Gan. Branch-specific dendritic  $Ca^{2+}$  spikes cause persistent synaptic plasticity. *Nature*, 520(7546):180–185, 2015.
- [11] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, S. Legg, and K. Kavukcuoglu. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. *arxiv*, 2018.
- [12] N. L. Etcoff and J. J. Magee. Categorical perception of facial expressions. *Cognition*, 44:227–240, 1992.
- [13] D. J. Freedman, M. Riesenhuber, T. Poggio, and E. K. Miller. Categorical representation of visual stimuli in the primate prefrontal cortex. *Science*, 291:312–316, 2001.
- [14] R. M. French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.
- [15] T. Furlanello, J. Zhao, A. M. Saxe, L. Itti, and B. S. Tjan. Active long term memory networks. *arXiv preprint arXiv:1606.02355*, 2016.
- [16] M. Garnelo, K. Arulkumaran, and M. Shanahan. Towards deep symbolic reinforcement learning. *arXiv preprint arXiv:1609.05518*, 2016.
- [17] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arxiv*, 2013.
- [18] P. D. Grünwald. *The minimum description length principle*. MIT press, 2007.
- [19] B. Gulyas, D. Ottoson, and P. E. Roland. *Functional Organisation of the Human Visual Cortex*. Wenner–Gren International Series, 1993.
- [20] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *ICCV*, 2015.
- [21] 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. *ICLR*, 2017.
- [22] I. Higgins, A. Pal, A. Rusu, L. Matthey, C. Burgess, A. Pritzel, M. Botvinick, C. Blundell, and A. Lerchner. DARLA: Improving zero-shot transfer in reinforcement learning. *ICML*, 2017.
- [23] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.
- [24] H. Kim and A. Mnih. Disentangling by factorising. *arxiv*, 2017.
- [25] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.
- [26] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *ICLR*, 2014.
- [27] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *PNAS*, 114(13):3521–3526, 2017.
- [28] A. Kumar, P. Sattigeri, and A. Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. *ICLR*, 2018.
- [29] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, pages 1–101, 2016.
- [30] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [31] Y. Liu and B. Jagadeesh. Neural selectivity in anterior inferotemporal cortex for morphed photographic images during behavioral classification or fixation. *J. Neurophysiol.*, 100:966–982, 2008.

- [32] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. *ICCV*, 2015.
- [33] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419, 1995.
- [34] M. McCloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *The psychology of learning and motivation*, 24(92):109–165, 1989.
- [35] K. Milan, J. Veness, J. Kirkpatrick, D. Hassabis, A. Koop, and M. Bowling. The forget-me-not process. *NIPS*, 2016.
- [36] V. Mnih, K. Kavukcuoglu, D. S. 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*, 518(7540):529–533, 2015.
- [37] L. Necker. Observations on some remarkable optical phenomena seen in switzerland; and on an optical phaenomenon which occurs on viewing a figure of a crystal or geometrical solid. *London and Edinburgh Philosophical Magazine and Journal of Science*, 1(5):329–337, 1832.
- [38] C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner. Variational continual learning. *ICLR*, 2018.
- [39] E. Parisotto, J. L. Ba, and R. Salakhutdinov. Actor-mimic: Deep multitask and transfer reinforcement learning. *ICLR*, 2015.
- [40] A. Przybyszewski. Vision: Does top-down processing help us to see? *Current Biology*, 8:135–139, 1998.
- [41] J. Ramapuram, M. Gregorova, and A. Kalousis. Lifelong generative modeling. *arXiv preprint arXiv:1705.09847*, 2017.
- [42] R. Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990.
- [43] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *ICML*, 32(2):1278–1286, 2014.
- [44] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.
- [45] A. Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, 1995.
- [46] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell. Progressive neural networks. *arxiv*, 2016.
- [47] P. Ruvolo and E. Eaton. Ella: An efficient lifelong learning algorithm. *ICML*, 2013.
- [48] A. Seff, A. Beatson, D. Suo, and H. Liu. Continual learning in generative adversarial nets. *NIPS*, 2017.
- [49] H. Shin, J. K. Lee, J. Kim, and J. Kim. Continual learning with deep generative replay. *NIPS*, 2017.
- [50] R. Shwartz-Ziv and N. Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.
- [51] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.
- [52] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arxiv*, 2017.
- [53] F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. *ICML*, 2017.

## A Supplemental Details

### A.1 Model details

**Encoder and decoder** For the encoder we use a simple convolutional network with the following structure:  $\text{conv } 64 \rightarrow \text{conv } 64 \rightarrow \text{conv } 128 \rightarrow \text{conv } 128 \rightarrow \text{fc } 256$ , where  $\text{conv } n\_filters$  is a  $4 \times 4$  convolution with  $n\_filters$  output filters, ReLU activations and stride 2, and similarly  $\text{fc } n\_out$  is a fully connected layer with  $n\_out$  units. The output of the fully connected layer is given to a linear layer that outputs the mean  $\mu_{\text{enc}}(\mathbf{x})$  and log-variance  $\log \sigma_{\text{enc}}^2(\mathbf{x})$  of the encoder posterior  $q_\phi(\mathbf{z}|\mathbf{x}) \sim N(\mu_{\text{enc}}(\mathbf{x}), \sigma_{\text{enc}}^2(\mathbf{x}))$ . The decoder network receives a sample  $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$  from the encoder and outputs the parameters of a distribution  $p_\theta(\mathbf{x}|\mathbf{z}, s)$  over  $\mathbf{x}$ . We use the transpose of the encoder network, but we also feed it the environment index  $s$  by first encoding it with a one-hot encoding (of size  $\text{max\_environments}$ , which is a hyperparameter), and then concatenating it to  $\mathbf{z}$ . For most of the experiments, we use a product of independent Bernoulli distributions (parametrised by the mean) for the decoder distribution  $p_\theta(\mathbf{x}|\mathbf{z}, s)$ . In the DM Lab experiments we use instead a product of Gaussian distributions with fixed variance. We train the model using Adam [25] with a fixed learning rate 6e-4 and batch size 64.

**Environment inference network** We attach an additional fully connected layer to the last layer of the encoder (gradients to the encoder are stopped). Given an input image  $\mathbf{x}$ , the layer outputs a softmax distribution  $q_\psi(s|\mathbf{x})$  over  $\text{max\_environments}$  indices, which tries to infer the most likely index  $s$  of the environment from which the data is coming, assuming the environment was already seen in the past. Notice that we always know the (putative) index of the current data ( $\hat{s}$  in Equation (6), also see Appendix A.2), so that we can always train this layer to associate the current data to the current index. However, to avoid catastrophic forgetting we also need to train on hallucinated data from past environments. Assuming  $\hat{s}$  is the current environment and  $m$  is the total number of environments seen until now, the resulting loss function is given by:

$$\mathcal{L}_{\text{env}} = \underbrace{\mathbb{E}_{\mathbf{x}}[-\log(q_\psi(\hat{s}|\mathbf{x}))]}_{\text{Classification loss on current data}} + \underbrace{\mathbb{E}_{\hat{s} \neq s < m} \mathbb{E}_{\mathbf{x}' \sim p_{\theta'}(\mathbf{x}'|\mathbf{z}', s)}[-\log q_\psi(s|\mathbf{x}')]}_{\text{Classification loss on hallucinated data}},$$

where the hallucinated data  $\mathbf{x}'$  in the second part of the equation is generated according to Section 3.5, and the expectation over  $s$  is similarly done through Monte Carlo sampling.

### A.2 Extra algorithm implementation details

**Atypical latent components** The atypicality  $\alpha_n$  of the component  $z_n$  on a batch of samples  $\mathbf{x}_1, \dots, \mathbf{x}_B$  is computed using a KL divergence from the marginal posterior over the batch to the prior according to Equation (4). In practice it is not convenient to compute this KL divergence directly. Rather, we observe that the marginal distribution of the latent samples  $\frac{1}{B} \sum_{b=1}^B q_\phi(z_n|\mathbf{x}_b)$  is approximately Gaussian. We exploit this by fitting a Gaussian to the latent samples  $\mathbf{z}$  and then computing in closed-form the KL-divergence between this approximation of the marginal and the unit Gaussian prior  $p(\mathbf{z}) = \mathcal{N}(0, 1)$ .

Recall from Section 3.3 that we deem a latent component  $z_n$  to be active ( $a_n = 1$ ) whenever it is typical, that is, if  $\alpha_n < \lambda$ . However, since the atypicality is computed on a relatively small batch of  $B$  samples,  $\alpha_i$  may be a noisy estimate of atypicality. Hence we introduce the following filtering: we set  $\alpha_n = 1$  if  $\alpha_n > \lambda_1$  and  $\alpha_n = 0$  if  $\alpha_n < \lambda_0$ , with  $\lambda_0 < \lambda_1$ . If  $\lambda_0 < \alpha_n < \lambda_1$ , we leave  $\alpha_n$  unchanged.

**Used latent components** We say that a factor  $z_n$  is not used by the environment  $s$  if the reconstruction  $p_\theta(\mathbf{x}|\mathbf{z}, s)$  does not depend on  $z_n$ . To measure this, we find the maximum amount of noise we can add to  $z_n$  without changing the reconstruction performance of the network. That is, we optimise

$$\Sigma = \underset{\Sigma = \text{diag}(\sigma_1, \dots, \sigma_N)}{\operatorname{argmin}} \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \sigma)}[-\log p_\theta(\mathbf{x}|\mathbf{z}^\epsilon, s)] - \log |\Sigma|$$

where  $\mathbf{z}^\epsilon_n = (1 - \delta_{nm})z_n + \delta_{nm}(z_n + \epsilon)$ . If  $\sigma'_n > T$  for some threshold  $T$ , we say that  $z_n$  is unused. We generally observe that components are either completely unused  $\sigma_n = 0$ , or else  $\sigma_n$  is very large. Therefore, picking a threshold is very easy and the precise value does not matter. We only compare the atypicality masks in eq. (6) for the used latents.

**Environment index** Expanding on the explanation in Section 3.4, let  $L_s(\mathbf{x}) = \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})}[-\log p_\theta(\mathbf{x}|\mathbf{z}^s, s)]$  be the reconstruction loss on a batch  $\mathbf{x}$  of data, assuming it comes from the environment  $s$ . Let  $\tilde{L}_s$  be the average reconstruction loss observed in the past for data from environment  $s$ . Let  $m$  be the number of datasets observed until now. Let  $\mathbf{u}^s$  be a binary vector of used units computed with the method described before.

We run the auxiliary environment inference network (Appendix A.1) on each sample from the batch  $\mathbf{x}$  and take the average of all results in order to obtain a probability distribution  $q(s|\mathbf{x})$  over the possible environment  $s$  of the batch  $\mathbf{x}$ , assuming it has already been seen in the past. Let  $\hat{s} = \operatorname{argmax}_s q(s|\mathbf{x})$  be the most likely environment, which is our candidate for the new environment. If the reconstruction loss  $L_{\hat{s}}(\mathbf{x})$  (assuming  $s = \hat{s}$ ) is significantly larger (see

EXPERIMENT	$\gamma$	$C_{\text{MAX}}$	DISENTANGLED			$\lambda$	$\kappa$	$\tau$	ENTANGLED		
			$\delta C$	$\lambda$	$\kappa$				$\lambda$	$\kappa$	$\tau$
ABLATION STUDY (150K)	100.0	35.0	6.3e-6	0.6	1.5	500					
FIVE DATASETS (150K)	100.0	35.0	6.3e-6	0.6	1.5	5000					
CELEBA $\rightarrow$ INVERTED FASHION (30K)	200.0	20.0	1.7e-5	0.8	1.1	500					
NATLAB $\rightarrow$ EDE (60K)	200.0	25.0	1e-5	2.0	1.1	5000					
IMAGINATION-DRIVEN EXPLORATION (45K)	200.0	35.0	0.7e-5	0.7	1.5	500			20.0	1.1	5000

Table 2: Hyperparameter values used for the experiments reported in this paper. Values in brackets after the experiment name indicate the number of training steps used per dataset.

Algorithm 1) than the average loss for the environment  $\hat{s}$ , we decide that the data is unlikely to come from this environment, and hence we allocate a new one. If the reconstruction is good, but some of the used components (given by  $\mathbf{u}$ ) are atypical, we still allocate a new environment. Otherwise, we assume that the data indeed comes from  $\hat{s}$ .

---

**Algorithm 1** Infer the environment index  $s$  from a batch of data

---

```

 $\hat{s} \leftarrow \operatorname{argmax}_s \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [-\log p_\theta(\mathbf{x}|\mathbf{z}^s, s)]$ 
if  $L_{\hat{s}} > \kappa \tilde{L}_{\hat{s}}$  then
     $s \leftarrow m+1$ 
else if  $a^{\hat{s}} \odot \mathbf{u}^s \neq \mathbf{a}(\mathbf{x}) \odot \mathbf{u}^{\hat{s}}$  then
     $s \leftarrow m+1$ 
else
     $s \leftarrow \hat{s}$ 

```

---

### A.3 Hyperparameter sensitivity

Table 2 lists the values of the hyperparameters used in the different experiments reported in this paper.

For all experiments we use `max_environments` = 7, and we increase  $C$  in eq. (3) linearly by  $\delta C \cdot C_{\text{max}}$  per step (starting from 0) until it reaches  $C_{\text{max}}$ , at which point we keep  $C$  fixed at that value. In the loss function eq. (8), the dreaming loss was re-weighted, with the full loss being:

$$\begin{aligned} \mathcal{L}(\phi, \theta) = & \underbrace{\mathbb{E}_{\mathbf{z}^s \sim q_\phi(\cdot|\mathbf{x}^s)} [-\log p_\theta(\mathbf{x}|\mathbf{z}^s, s)] + \gamma |\mathbb{KL}(q_\phi(\mathbf{z}^s|\mathbf{x}^s)||p(\mathbf{z})) - C|^2}_{\text{MDL on current data}} \\ & + \underbrace{\mathbb{E}_{\mathbf{z}, s', \mathbf{x}'} \left[ \alpha D[q_\phi(\mathbf{z}|\mathbf{x}'), q_{\phi'}(\mathbf{z}'|\mathbf{x}')] + \beta D[q_\theta(\mathbf{x}|\mathbf{z}, s'), q_{\theta'}(\mathbf{x}'|\mathbf{z}, s')] \right]}_{\text{"Dreaming" feedback on past data}}. \end{aligned} \quad (9)$$

The values  $\alpha = 1000$  and  $\beta = 20$  were used for all experiments, except for in the hyperparameter sweep.

For the ablation study we ran a hyperparameter search using the full model, and used the best hyperparameters found for all experiments. We list the search ranges and our observed sensitivity to these hyperparameters next:

- $\gamma$  = coefficient for the capacity constraint – {50,100,200} – found not to be very sensitive.
- $C_{\text{max}}$  = final value of  $C$  – {20,35,50} – classification accuracy increased significantly for capacity from 20 to 35.
- $\lambda$  = atypicality threshold – {0.4,0.6,1,2} – lower threshold led to more latent freezing. Classification performance was not very sensitive to this.
- $\tau$  = update frequency reference network – {500,1000,2000,5000} – found not to be very sensitive.
- $\alpha$  = weight for encoder loss in "dreaming" loop – {10,20,40} – found not to be very sensitive.
- $\beta$  = weight for decoder loss in "dreaming" loop – {500,1000,2000} – found not to be very sensitive.

#### A.3.1 Imagination-based exploration experiments

We model a very simple interaction with the environment where the agent can translate the observed object (in our case the input image). The agent is trained as follows: a random  $\mathbf{z}^*$  is sampled from the prior  $p(\mathbf{z})$ . Given an observation  $\mathbf{x}$  from the environment, the agent needs to pick an action  $g(\mathbf{z}^*, \mathbf{x})$  (in our case a translation) in such a way that the encoding  $\mathbf{z} \sim q_\phi(\mathbf{z}|g \cdot \mathbf{x})$  of the new image  $g \cdot \mathbf{x}$  is as close as possible to  $\mathbf{z}^*$ . That is, we minimise the loss

$$\mathcal{L}_{\text{agent}} = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \mathbb{E}_{\mathbf{z}^* \sim p(\mathbf{z})} \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|g(\mathbf{z}^*, \mathbf{x}))} \|\mathbf{z}^* - \mathbf{z}\|^2.$$

The agent can then be used to explore the current environment. Given an image  $\mathbf{x}$  from the current environment, we can imagine a configuration  $\mathbf{z}^*$  of the latent factors, and let the agent act on the environment in order to realise the configuration. In this way, we obtain a new image  $\mathbf{x}^* = g(\mathbf{z}, \mathbf{x}) \cdot \mathbf{x}$ . We can then add the image  $\mathbf{x}^*$  to the training data, which allows the encoder to learn from a possibly more diverse set of inputs than the inputs  $\mathbf{x} \sim p(\mathbf{x})$  observed passively.

The policy network first processes the input image  $\mathbf{x}$  (of size  $64 \times 64$ ) through four convolutional  $4 \times 4$  layers with 16 filters, stride 2 and ReLU activations. The resulting vector is concatenated with the target  $\mathbf{z}^*$ , and feed to a 1-hidden layer fully connected network that outputs the parameters of the 2D translation  $g(\mathbf{x}, \mathbf{z}^*)$  to apply to the image. We use a  $\tanh$  output to ensure the translation is always in a sensible range. Once these parameters are obtained, the transformation is applied to the image  $x$  using a Spatial Transformer Network (STN) [23], obtaining a translated image  $\mathbf{x}^* = g(\mathbf{z}^*, \mathbf{x}) \cdot \mathbf{x}$ . We can now finally compute the resulting representation  $\mathbf{z} \sim q(\mathbf{z} | g(\mathbf{z}^*, \mathbf{x}) \cdot \mathbf{x})$ . Notice that the whole operation is fully differentiable, thanks to the properties of the STN. The policy is can now be trained by minimising  $\mathcal{L}_{\text{agent}}$  in order to make  $\mathbf{z}$  and  $\mathbf{z}^*$  as close as possible. In our experiments we train the policy while training the main model.

#### A.4 Dataset processing

**DM Lab** We used an IMPALA agent trained on all DM-30 tasks [11] to generate data. We take observations of this optimal agent (collecting rewards according to the task descriptions explained in [https://github.com/deepmind/lab/tree/master/game\\_scripts/levels/contributed/dmlab30](https://github.com/deepmind/lab/tree/master/game_scripts/levels/contributed/dmlab30)), on randomly generated episodes of Exploit Deferred Effects and NatLab Varying Map Randomized; storing them as  $111 \times 84 \times 3$  RGB tensors. We crop the right-most 27 pixels out to obtain a  $84 \times 84$  image (this retains the most useful portion of the original view), which are finally scaled down to  $64 \times 64$  (using `tf.image.resize_area`).

**CelebA → Inverse Fashion** To make CelebA compatible with Fashion, we convert the CelebA images to grayscale and extract a patch of size  $32 \times 32$  centered on the face. We also invert the colours of Fashion so that the images are black on a white background, and slightly reducing the contrast, in order to make the two datasets more similar, and hence easier to confuse after mixing.

#### A.5 Quantifying catastrophic forgetting

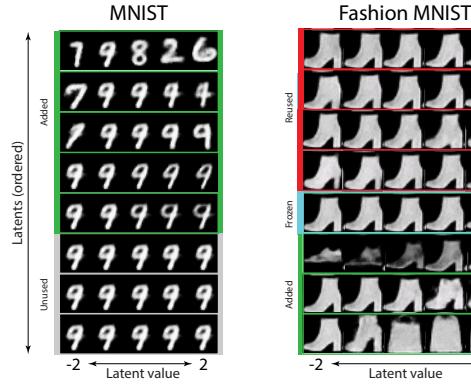
We train on top of the representation  $\mathbf{z} \sim q_\phi(\mathbf{z} | \mathbf{x})$  a simple 2-hidden layers fully connected classifier with 256 hidden units per layer and ReLU activations. At each step while training the representation, we also train a separate classifier on the representation for each environment, using Adam with learning rate 6e-4 and batch size 64. This classifier training step does not update the weights in the main network.

For each ablation type we reported the average classification accuracy (or regression MSE) score obtained by 20 replicas of the model, all with the best set of hyperparameters discovered for the full model. We quantified catastrophic forgetting by reporting the average difference between the maximum accuracy obtained while VASE was training on a particular dataset and the minimum accuracy obtained for the dataset afterwards.

#### A.6 Additional results

We present additional experimental results and extra plots for the experiments reported in the main paper here. Fig. 5 and table 3 show latent traversals and quantitative evaluation results for an ablation study on VASE trained on the MNIST → Fashion → MNIST sequence. Fig. 5 also shows traversals for VASE trained on the DM Lab levels NatLab → EDE. This is the model reported in section 4. Fig. 6 shows cross-dataset reconstructions for VASE trained on the moving Fashion → MNIST → moving MNIST sequence described in the ablation study in section 4. Figs. 7-8 shows latent traversals and cross-dataset reconstructions for VASE trained on the moving MNIST → Fashion → inverted Fashion → MNIST → moving Fashion sequence described in the main text.

## MNIST Traversals



## DM Lab Traversals

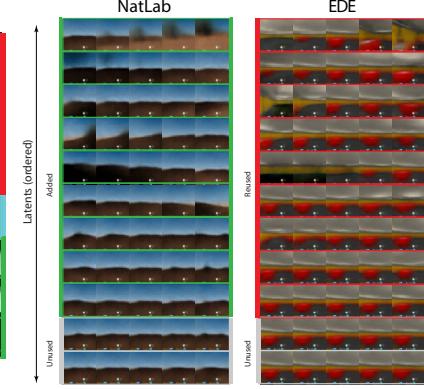


Figure 5: Latent traversals for VASE trained on MNIST → Fashion → MNIST, and DM Lab levels NatLab → EDE.

## Moving Fashion/Moving MNIST Cross-Domain Reconstructions

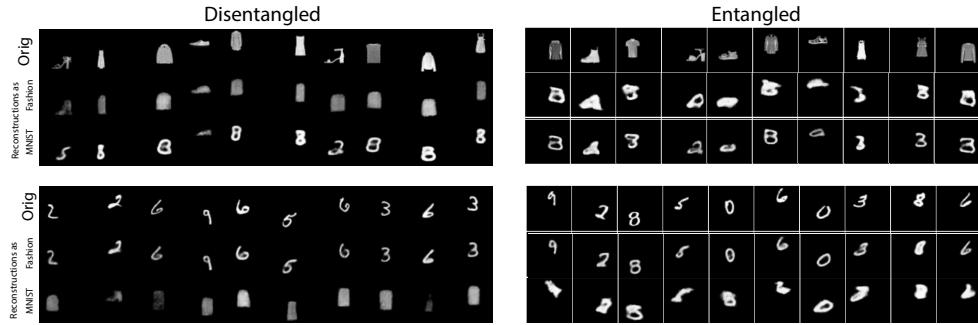


Figure 6: Cross-domain reconstructions for the entangled and disentangled versions of VASE (as described in section 4) trained on moving Fashion → MNIST → moving MNIST. We see that the entangled baseline forgets moving Fashion by the end of training.

## Latent Traversals

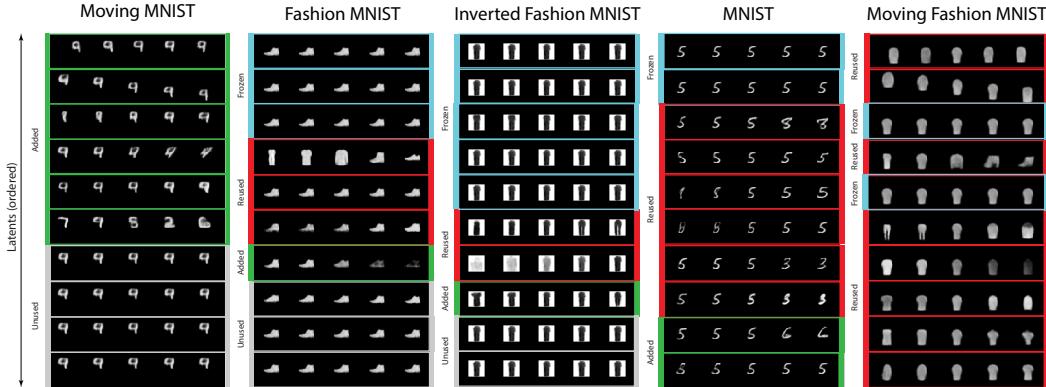


Figure 7: Latent traversals for VASE trained on a sequence of moving MNIST → Fashion → inverse Fashion → MNIST → moving Fashion.

CONFIGURATION	DISENTANGLED		ENTANGLED	
	AVG. DECREASE (%)	Avg. Max (%)	Avg. Decrease (%)	Avg. Max (%)
DA	-7.9	90.5	-12.1	90.9
SD	-2.2	91.0	-4.3	<b>92.1</b>
S	-3.9	90.8	-8.4	91.5
A	-9.6	90.2	-10.1	91.7
SA	-5.9	90.0	-10.3	91.0
-	-4.4	91.1	-6.6	92.6
D	-6.0	90.5	-6.9	91.4
<b>VASE (SDA)</b>	<b>-0.9</b>	90.3	-2.5	91.2

Table 3: Average drop in classification accuracy and maximum average accuracy when training an object classifier on top of the learnt representation on the MNIST → Fashion → MNIST sequence. We do a full ablation study of VASE, where D - dreaming feedback loop, S - cluster inference  $q(s|x^s)$ , and A - atypicality based latent mask  $\alpha^s$  inference. We compare two versions of our model - one that is encouraged to learn a disentangled representation through the capacity increase regularisation in eq. (3), and an entangled VAE baseline ( $\beta = 1$ ). The unablated disentangled version of VASE (SDA) has the best performance.

## Cross-Domain Reconstructions

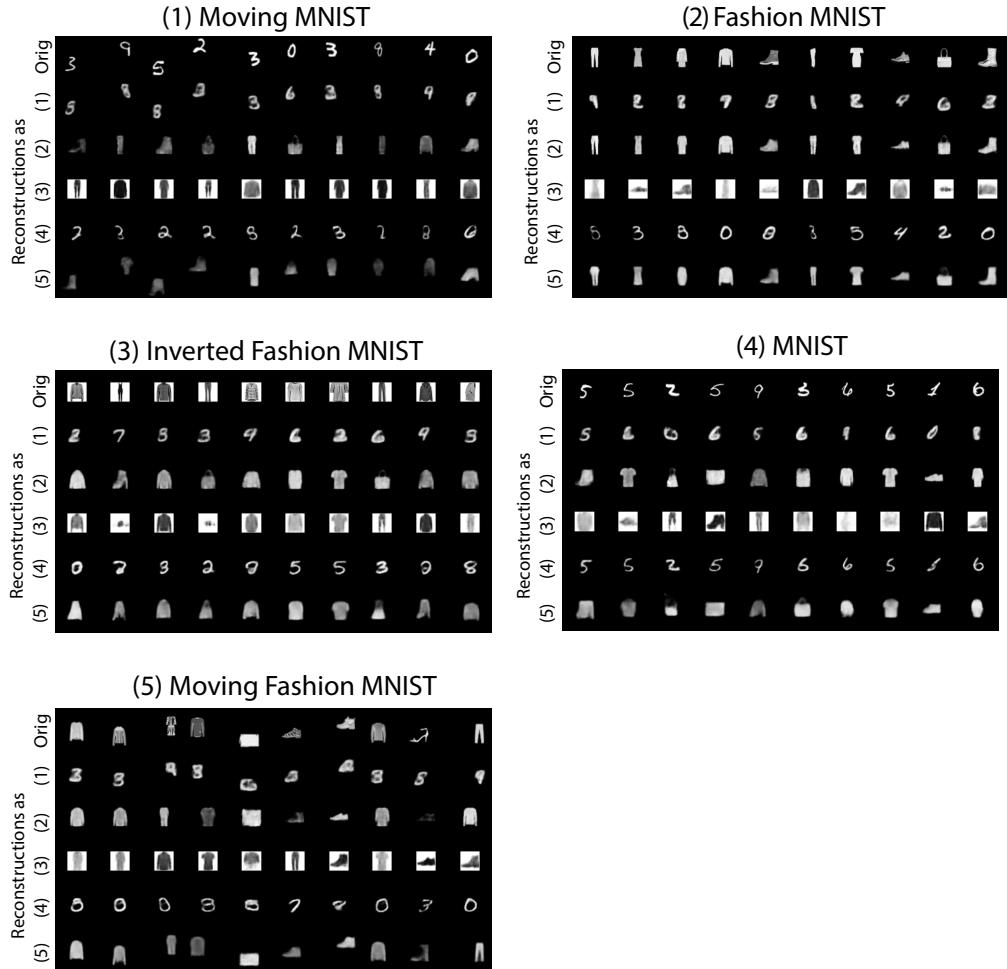


Figure 8: Cross-domain reconstructions for the VASE trained on moving MNIST → Fashion → inverted Fashion → MNIST → moving Fashion.