A Extended background

We use SVEA (Hansen et al., 2021) as the base RL algorithm for the CMID auxiliary task, which is an extension of the Soft Actor-Critic (SAC) algorithm (Haarnoja et al., 2018).

SAC is an off-policy RL algorithm for continuous control. SAC learns a stochastic policy π that maximises the expected sum of rewards and the entropy of the policy. The critic Q is learned by minimising the loss:

$$L_Q = \mathbb{E}_{(\mathbf{o}_t, \mathbf{a}_t, \mathbf{o}_{t+1}, r_t) \sim \mathcal{D}} \left[\left(Q(\mathbf{o}_t, \mathbf{a}_t) - r_t - \gamma \bar{V}(\mathbf{o}_{t+1})) \right)^2 \right] \tag{8}$$

where $\mathbf{o}_t$ is the image observation and $\mathbf{a}_t$ is the action at time t as defined in Section 3. SAC uses the minimum of two Q networks, Q_1 and Q_2, for the training updates to reduce overestimation of Q values. The actor π is trained by minimising the loss:

$$L_\pi = \mathbb{E}_{\mathbf{o}_t \sim \mathcal{D}} \left[\mathbb{E}_{\mathbf{a}_t \sim \pi} \left[\alpha_{\text{SAC}} \log(\pi(\mathbf{a}_t \mid \mathbf{o}_t)) - \min_{i=1,2} \bar{Q}_i(\mathbf{o}_t, \mathbf{a}_t) \right] \right] \tag{9}$$

where $\bar{Q}$ is exponential moving average of the Q network parameters.

SVEA aims to stabilise SAC training using a combination of both augmented and unaugmented images for Q learning with an modified loss:

$$L_Q^{\text{SVEA}} = \alpha_{\text{SVEA}} L_Q(o_t, a_t, o_{t+1}) + \beta_{\text{SVEA}} L_Q(o_t^{\text{aug}}, a_t, o_{t+1}) \tag{10}$$

However, the actor π is optimised on unaugmented images only, using the SAC policy loss in Equation 9.

B Implementation details

In this section, we provide the implementation details for CMID. Our codebase is built on top of the publicly released DrQ PyTorch implementation by Yarats et al. (2021) as well as the official implementation of SVEA by Hansen et al. (2021). A public and open-source implementation of CMID is available at github.com/usr/repo [currently anonymised for double blind review].

Encoder. The encoder consists of 4 convolutional layers, each with a 3×3 kernel size and 32 channels. The first layer has a stride of 2, all other layers have a stride of 1. There is a ReLU activation between each of the convolutional layers. The convolutional layers are followed by a linear layer, normalisation, then a tanh activation. The encoder weights are shared between the actor π and critic Q.

Actor and critic. Both the actor π and critic Q networks are MLPs consisting of two layers and a hidden dimension of 1024. There is a ReLU activation after each layer except the last layer.

CMID discriminator. The CMID discriminator is implemented as an MLP consisting of two layers and a hidden dimension of 1024. There is a ReLU activation after each layer except the last layer. The same conditional discriminator is used for all features in the representation so the inputs are one-hot encoded. This means the input size is: 56 (representation or permuted representation) + 56 (one-hot encoding of previous representation) + action size.

Hyperparameters. We tuned learning rate and CMID hyperparameters by grid search; other hyperparameters follow the original SVEA implementation. Table 2 shows the hyperparameters for all tasks.

Hardware. For each experiment run we use a single NVIDIA Volta V100 GPU with 32GB memory and a single CPU.

Hyperparameter	Value
Replay buffer capacity	100000
Initial steps before training begins	1000
Stacked frames (stacked representations for CMID)	3
Action repeat	2 for finger_spin, 8 for cartpole_swingup, 4 otherwise
Batch size	128
Discount factor	0.99
Optimizer	Adam
Learning rate (actor, critic and encoder)	1e-3
SAC learning rate for α_{SAC}	1e-4
Discriminator learning rate (CMID only)	1e-2
SVEA coefficients	$\alpha_{\text{SVEA}} = 0.5$, $\beta_{\text{SVEA}} = 0.5$
Target soft-update rate τ	critic 0.01, actor 0.05
Actor update frequency	2
Actor log stddev bounds	$[-10, 2]$
Latent representation dimension	56
Image size	(84, 84, 3)
Image pad	4
Initial temperature	0.1
CMID loss coef α	0.5 for cartpole_swingup, 0.1 otherwise
k nearest neighbours	5

Table 2: Hyperparameter values for both SVEA and SVEA-CMID.

	cartpole_swingup	walker_walk	finger_spin	hopper_stand
SVEA-CMID	$\mathbf{746.0 \pm 77.8}$	$\mathbf{793.5 \pm 36.0}$	$\mathbf{939.5 \pm 19.1}$	$\mathbf{826.0 \pm 15.6}$
SVEA	233.1 ± 25.3	460.8 ± 50.7	633.8 ± 122.6	686.3 ± 170.8
SVEA-TED	577.0 ± 152.0	542.7 ± 115.1	755.4 ± 75.8	623.5 ± 166.7
CURL	262.4 ± 34.8	285.7 ± 54.3	386.3 ± 141.5	305.6 ± 160.4
DrQ	201.2 ± 20.7	417.3 ± 32.1	843.3 ± 49.1	531.8 ± 182.6

Table 3: Zero-shot generalisation performance to *reversed correlation*. Returns are the average of 10 evaluation episodes over 5 seeds, showing ± standard error.

C Additional results

C.1 Zero-shot generalisation

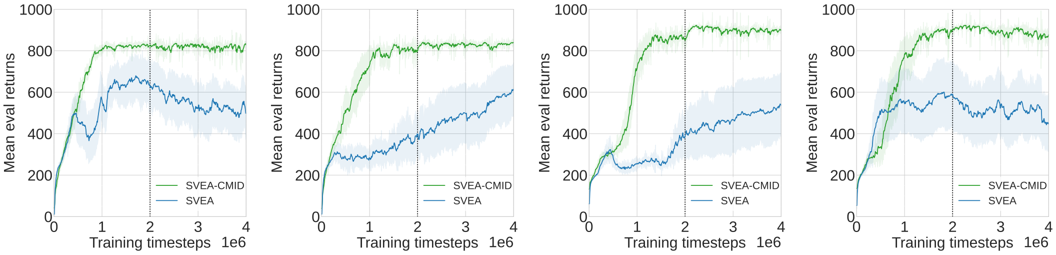
The zero-shot generalisation performance under correlation shift can be seen at the vertical dotted line in the graphs of Figure 4 and Figure 5. For completeness and to avoid loss of information caused by smoothing in the graphs, the numerical values of the zero-shot generalisation performance are provided in Table 3 and Table 4.

C.2 Evaluation on each scenario

The results in Section 5 show the average returns over 10 evaluation episode for each seed, where a given scenario is selected based on the train/test probabilities depicted in Figure 3. To further assess performance, Figure 10 shows the average evaluation returns on 10 episodes for each object/colour combination on the cartpole swingup task with generalisation to reversed correlation. These results show that the correlation makes it difficult for SVEA to learn an optimal policy for any scenario, but with lower returns on the unlikely training scenarios in particular (cartpole A in green and cartpole B in blue). This explains the failure to generalise in Figure 4 when the correlation reverses, making the scenarios that were rare in training become frequent in testing at the vertical dotted line.

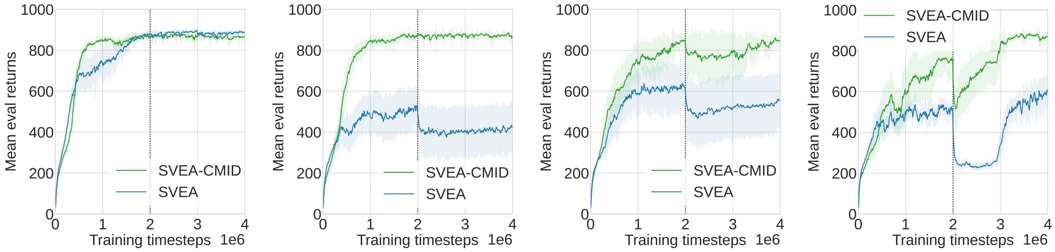
	cartpole_swingup	walker_walk	finger_spin	hopper_stand
SVEA-CMID	**878.8 ± 12.4**	**815.3 ± 29.9**	**953.2 ± 16.4**	**816.1 ± 37.2**
SVEA	371.5 ± 21.0	652.4 ± 34.3	680.1 ± 98.4	526.8 ± 182.2
SVEA-TED	667.4 ± 120.6	560.7 ± 68.6	820.7 ± 59.6	643.3 ± 171.0
CURL	523.8 ± 83.6	606.3 ± 50.8	561.7 ± 119.9	342.9 ± 151.1
DrQ	521.6 ± 55.3	652.5 ± 26.4	872.0 ± 30.3	531.6 ± 149.5

Table 4: Zero-shot generalisation performance to *uncorrelated* variables. Returns are the average of 10 evaluation episodes over 5 seeds, showing ± standard error.



(a) Cartpole A in blue **(b)** Cartpole A in green **(c)** Cartpole B in blue **(d)** Cartpole B in green

Figure 10: Evaluation of performance on each of the cartpole swingup scenarios for generalisation to reversed correlation, averaged over 10 evaluation episodes for 5 seeds.



(a) correlation = 0.7 **(b)** correlation = 0.8 **(c)** correlation = 0.9 **(d)** correlation = 0.99

Figure 11: Generalisation to reversed correlation at the vertical dotted line with varying correlation strengths on the cartpole swingup task.

C.3 Correlation strength.

The generalisation results in Section 5 show training with a 0.95 correlation (0.95 probability of being on the leading diagonal in Figure 3, and only 0.05 probability of being in the anti-diagonal scenarios). We conducted further analysis of different correlation strengths, denoting the sum of probabilities on the leading diagonal as the correlation strength. The results for generalisation to the reversed correlation are shown in Figure 11. While the generalisation performance of SVEA decreases as the correlation gets stronger, SVEA-CMID consistently generalises well up to a very strong correlation of 0.99 at which point the performance deteriorates but still significantly improves the performance of SVEA in this setting.

C.4 Greyscale images.

Our experiments use colour correlations to demonstrate the failure to generalise under correlation shifts. So we also demonstrate that the results still hold in greyscale images in Figure 12.

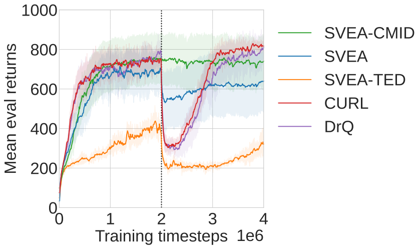
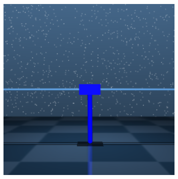
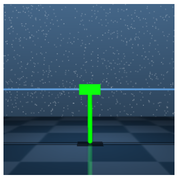
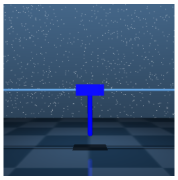
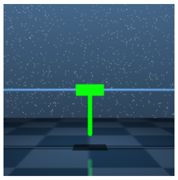
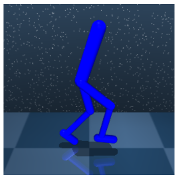
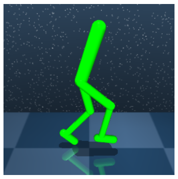
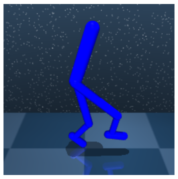
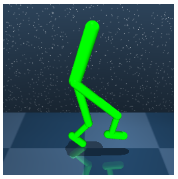
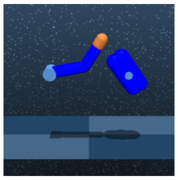
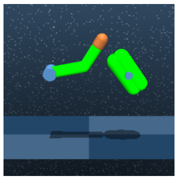
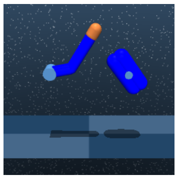
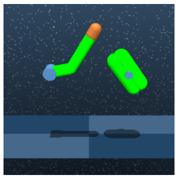


Figure 12: Generalisation to reversed correlation at the vertical dotted line on the cartpole swingup task with all image observations converted to greyscale.

D Environment variations

In Table 5, we provide a description of the differences between the two object variations (A and B) in each task, along with images of example observations for each object and colour combination. The exact specification of the world model for each task is available in our code.

Environment	Variation	Blue	Green
cartpole_swingup	A - original cartpole		
	B - wider cart, shorter pole		
walker_walk	A - original walker		
	B - longer thigh, shorter calf		
finger_spin	A - original finger		
	B - shorter proximal, longer distal		

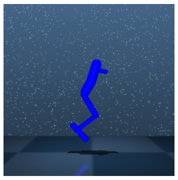
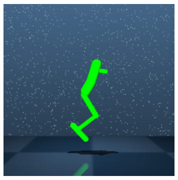
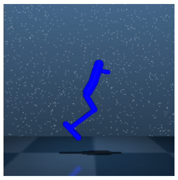
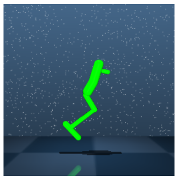
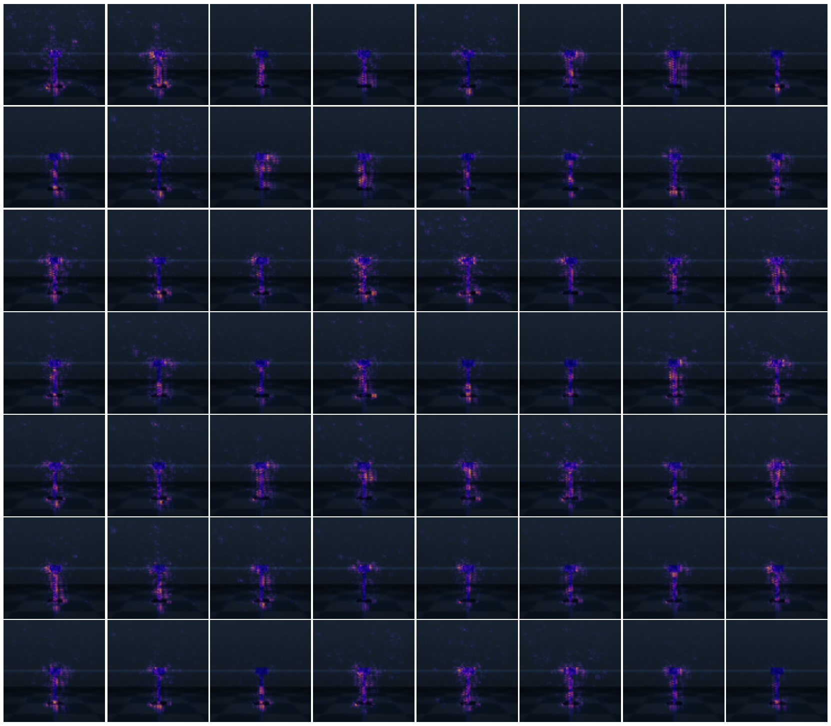
hopper_stand	A - original hopper		
	B - shorter thigh, longer calf		

Table 5: Environment images

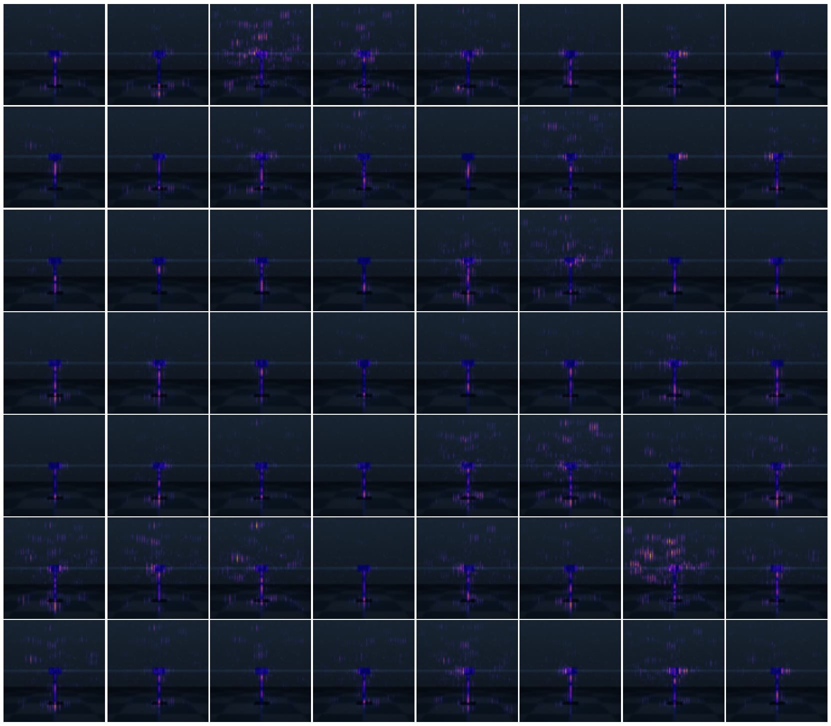
E Saliency maps

The full set of saliency maps, as described in Section 6, for each representation feature is provided in Figure 13 for a trained SVEA encoder and a trained SVEA-CMID encoder. The features are sorted in order of most active to least active based on the sum of attributions for each feature.

To create the saliency maps, we use the Captum open-source interpretability library for PyTorch (Kokhlikyan et al., 2020) to calculate the integrated gradients (Sundararajan et al., 2017) pixel attributions for each feature in the representation output of the encoder. We use an all black image as the baseline image for integrated gradients which is compared to the input image in Figure 9a. The absolute value of the attributions are overlayed onto the input image to create the saliency maps.



(a) SVEA



(b) SVEA-CMID

Figure 13: Saliency maps for each representation feature of a trained (a) SVEA and (b) SVEA-CMID encoder on the cartpole swingup task, sorted in order of highest total attributions to lowest.