Appendix

A. Network Architecture and Training Procedure

In this section, we elaborate more on our network architectures and training procedures. The algorithm for generation state and observation predictions is presented in Algorithm 1. The general algorithm for encoding and decoding both image and time-series data is presented in Algorithms 2-5. The temporal downsample block follows the implementation of WaveNet [1] (i.e. gated, dilated, causal convolutions). However, as there is no temporal order to the latent code, temporal upsampling is handled simply by 1D convolution and upsampling along the time dimension. We present the full list of neural network architectures in Tables I-VI. We present our training hyperparameters in Table VIII. Since we evaluate multiple different loss types, we add an additional column denoting which experiments used which hyperparameters (with 'R' standing for reconstruction and 'C' for contrastive).

B. T-SNE figures For Dynamical Variation Experiment

The full set of t-SNE figures and clusters from our motivational experiment are provided in Figures **??** and **??**, respectively. The hyperparameters for the experiment are provided in Table IX.

References

[1] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[2] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets. *arXiv preprint arXiv:1703.06114*, 2017.

[3] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Algorithm 1: Latent Model Forward Pass

Input: Modality set M,
initial state x_0, initial observations $\{o_0^m, \forall m \in M\}$, action sequence $a_{1:T}$, modality prediction set $\tilde{M}$. Encoders $e_\psi^m, \forall m \in M$, Decoders $d_\psi^m, \forall m\tilde{M}$, latent model $f_\theta(z, a)$, action encoder $g_\psi(a)$, state decoder d_ψ^{state}
Output: State predictions $x\tilde{_{1:T}}$, observation predictions $\{o_{1:t}^m, \forall m \in \tilde{M}\}$
for $m \in M$ **do**
    $p^m(z) \leftarrow e_\psi^m(o_0^m)$ ◁ Encode each observation into $\mathcal{Z}$
end
$z_0 = \text{aggregate}(\{p^m(z), \forall m \in M\})$ ◁ Use Deepsets [2] or Product of Experts [3] to get single z
for $t \in 1 : T$ **do**
    $a_{t-1} = g_\psi(at-1)$ ◁ *Embed action*
    $z_t = f_\theta(z, a_{t-1})$ ◁ *Predict next latent state*
    $x_t = d_\psi^{state}(z_t)$ ◁ *Decode state*
    for $m \in \tilde{M}$ **do**
        $o_{t+1}^m = d_\psi(z_t)$ ◁ *Decode observation*
    end
end
return $x_{1:T}, \{o_{1:t}^m, \forall m \in \tilde{M}\}$

Algorithm 2: Upsample Block

Input: Image input x, upsample factor s, convolution kernel K, activation function f
Output: Upsampled image output $\tilde{x}$
$x \leftarrow$ linear interpolate(x, s)
$x \leftarrow x * K$
$x \leftarrow f(x)$
return x

Algorithm 3: Downsample Block

Input: Image input x, downsample factor s, convolution kernel K, activation function f
Output: Downsampled image output $\tilde{x}$
$x \leftarrow x * K$
$x \leftarrow f(x)$
$x \leftarrow$ linear interpolate(x, s)
return x

Algorithm 4: CNN Encoder

Input: Image input x, downsample blocks D_ψ, MLP f_θ
Output: Latent distribution $p(z)$
for d_ψ *in D* **do**
    $x \leftarrow d_\psi(x)$ ◁ using Algorithm 3 or [1]
end
$x \leftarrow$ flatten(x) ◁ Flatten x to 1D
$\mu, \sigma \leftarrow f_\theta(x)$
return $\mathcal{N}(\mu, \sigma)$

Algorithm 5: CNN Decoder

Input: Latent vector z, upsample blocks U_ψ, MLP f_θ
Output: Image reconstruction $\tilde{X}$
$x \leftarrow f_\theta(x)$
$x \leftarrow$ pad_front $(x, 2)$ ◁ $x \in \{1 \times 1 \times |x|\}$
for u_ψ *in U* **do**
    $x \leftarrow u_\psi(x)$ ◁ using Algorithm 2
end
return x

Layer	Input Dim	Output Dim	Kernel Size	Activation
Downsample 1	$3 \times 128 \times 128$	$4 \times 64 \times 64$	3×3	ReLU
Downsample 2	$4 \times 64 \times 64$	$8 \times 32 \times 32$	3×3	ReLU
Downsample 3	$8 \times 32 \times 32$	$16 \times 16 \times 16$	3×3	ReLU
Downsample 4	$16 \times 16 \times 16$	$32 \times 8 \times 8$	3×3	ReLU
Flatten	$32 \times 8 \times 8$	2048	-	-
MLP	2048	$2 \times \|\mathcal{Z}\|$	-	Tanh
Gaussian	$2 \times \|\mathcal{Z}\|$	$\mathcal{N} \in \mathcal{Z}$	-	-

TABLE I
VISUAL CNN ENCODER ARCHITECTURE

Layer	Input Dim	Output Dim	Kernel Size	Activation
Downsample 1	$\{1, 3\} \times 64 \times 64$	$4 \times 32 \times 32$	3×3	ReLU
Downsample 2	$4 \times 32 \times 32$	$8 \times 16 \times 16$	3×3	ReLU
Downsample 3	$8 \times 16 \times 16$	$16 \times 8 \times 8$	3×3	ReLU
Downsample 4	$16 \times 8 \times 8$	$32 \times 4 \times 4$	3×3	ReLU
Flatten	$32 \times 4 \times 4$	512	-	-
MLP	512	$2 \times \|\mathcal{Z}\|$	-	Tanh
Gaussian	$2 \times \|\mathcal{Z}\|$	$\mathcal{N} \in \mathcal{Z}$	-	-

TABLE II
LOCAL MAP CNN ENCODER ARCHITECTURE

Layer	Input Dim	Output Dim	Kernel Size	Kernel Dilation	Activation
Downsample 1	$\{4,9\} \times 20$	$\{4,9\} \times 20$	2	2	[1]
Downsample 2	$\{4,9\} \times 20$	$\{4,9\} \times 20$	2	4	[1]
Downsample 3	$\{4,9\} \times 20$	$\{4,9\} \times 20$	2	8	[1]
Downsample 4	$\{4,9\} \times 20$	$\{4,9\} \times 20$	2	16	[1]
Flatten	$\{4,9\} \times 20$	$\{80,180\}$	-	-	-
MLP	$\{80,180\}$	$2 \times \lvert\mathcal{Z}\rvert$	-	-	Tanh
Gaussian	$2 \times \lvert\mathcal{Z}\rvert$	$\mathcal{N} \in \mathcal{Z}$	-	-	

TABLE III

TEMPORAL CNN ENCODER ARCHITECTURE

Layer	Input Dim	Output Dim	Kernel Size	Activation
MLP	$\lvert\mathcal{Z}\rvert$	128	-	Tanh
Unflatten	128	$128 \times 1 \times 1$	-	-
Upsample 1	$128 \times 1 \times 1$	$32 \times 4 \times 4$	3×3	ReLU
Upsample 2	$32 \times 4 \times 4$	$16 \times 8 \times 8$	3×3	ReLU
Upsample 3	$16 \times 8 \times 8$	$8 \times 16 \times 16$	3×3	ReLU
Upsample 4	$8 \times 16 \times 16$	$4 \times 32 \times 32$	3×3	ReLU
Upsample 5	$4 \times 32 \times 32$	$3 \times 128 \times 128$	3×3	ReLU

TABLE IV

VISUAL CNN DECODER ARCHITECTURE

Layer	Input Dim	Output Dim	Kernel Size	Activation
MLP	$\lvert\mathcal{Z}\rvert$	128	-	Tanh
Unflatten	128	$128 \times 1 \times 1$	-	-
Upsample 1	$128 \times 1 \times 1$	$32 \times 4 \times 4$	3×3	ReLU
Upsample 2	$32 \times 4 \times 4$	$16 \times 8 \times 8$	3×3	ReLU
Upsample 3	$16 \times 8 \times 8$	$8 \times 16 \times 16$	3×3	ReLU
Upsample 4	$8 \times 16 \times 16$	$4 \times 32 \times 32$	3×3	ReLU
Upsample 5	$4 \times 32 \times 32$	$3 \times 64 \times 64$	3×3	ReLU

TABLE V

LOCAL MAP CNN DECODER ARCHITECTURE

Layer	Input Dim	Output Dim	Kernel Size	Activation
Unflatten	$\lvert\mathcal{Z}\rvert$	$1 \times \lvert\mathcal{Z}\rvert$	-	-
Upsample 1	$1 \times \lvert\mathcal{Z}\rvert$	2×64	2	Tanh
Upsample 1	2×64	4×32	2	Tanh
Upsample 1	4×32	8×16	2	Tanh
Upsample 1	8×16	16×8	2	Tanh
Upsample 1	16×8	$20 \times \{4,9\}$	2	Tanh

TABLE VI

TEMPORAL CNN DECODER ARCHITECTURE

Layer	Input Dim	Output Dim	Activation
Action Encode 1	2	16	Tanh
Action Encode 2	2	16	Tanh
GRU	$(128, 23)$	$128, 128$	-
State Decode 1	128	128	Tanh
State Decode 2	128	$\mathcal{N} \in R^7$	-

TABLE VII

LATENT MODEL ARCHITECTURE

Hyperparameter	Value	Experiment
Optimizer	Adam [4]	All
Learning Rate	$1e-3$	All
Epochs	5000	All
Batch Size	64	All
Gradient Steps Per Epoch	10	All
Gradient Norm Clip	100.0	All
Train Timesteps	20	All
RGB Image Loss Scale	100	R
RGB Map Loss Scale	100	R
Heightmap Loss Scale	1	R
IMU Loss Scale	0.1	R
Wheel RPM Loss Scale	0.1	R
Contrastive Scale	10.0	C
EMA τ	0.05	C

TABLE VIII

TRAINING HYPERPARAMETERS

Hyperparameter	Value
# Subsequences	10000
Sequence length	10
# Clusters	10
# Velocity Bins	5
Clustering Distance Metric	Euclidean

TABLE IX

MOTIVATIONAL EXPERIMENT HYPERPARAMETERS