Heteroscedastic Uncertainty for Robust Generative Latent Dynamics

Supplementary Material.

Abstract

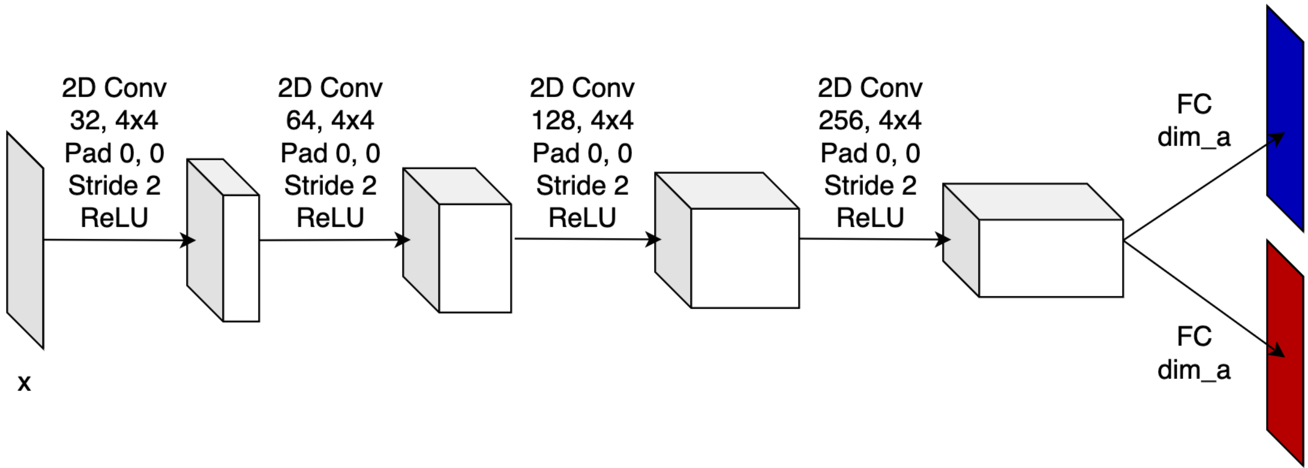
Supplementary material for the IROS 2020 paper *Heteroscedastic Uncertainty for Robust Generative Latent Dynamics*. We present additional results and implementation details from our experiments.

I. Network Architecture and Training Details

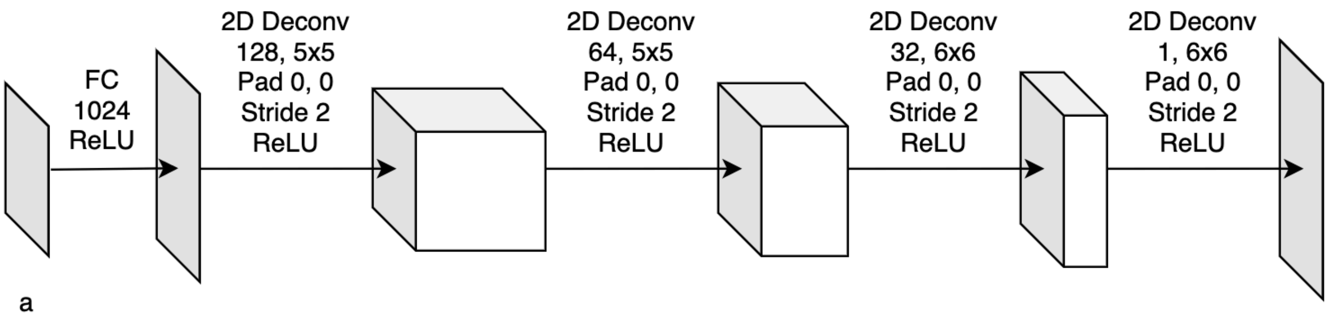
Task	Pendulum	Reacher
Trajectory Length (K)	32	15
Number of Trajectories	2048	700
Training Trajectory Length (I)	16	7
Amount of Base Matrices [1], [2]	15	15
Learning Rate	$3e^{-4}$	$3e^{-4}$
Training Epochs	4096	4096
Batch Size	64	64
Reconstruction Term Coefficient (λ_{REC})	0.95	0.95
KL Term Coefficient (λ_{KL})	0.8	0.8

TABLE I: Hyperparameters for training the model for the pendulum and real-world visual reaching task.

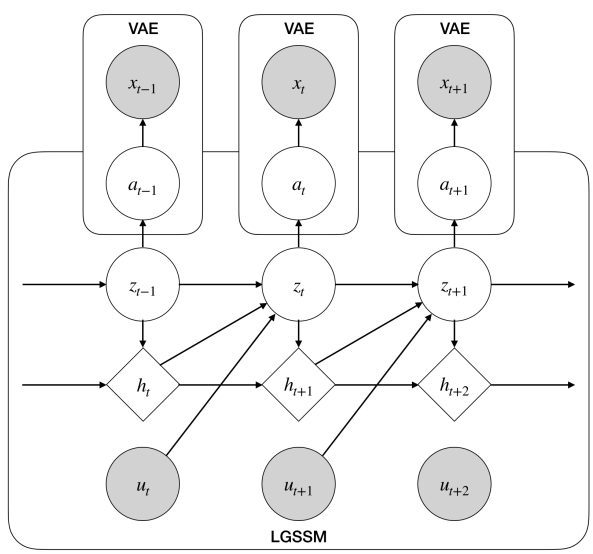
We provide a summary of the training hyperparameters used in Table I. We use the Adam optimizer [3] for both tasks. We clip the gradient norms to be 0.5 for each of the encoder, decoder, and LGSSM (along with the GRU network). Fully connected layers are initialized with Xavier Uniform Initialization [4]. Convolutional layers are initialized with Kaiming Uniform Initialization [5]. The GRU's hidden-to-hidden parameters are initialized with Orthogonal initialization [6] and input-to-hidden parameters are initialized with Xavier Uniform Initialization. All biases are initialized to zero. We first jointly train the VAE and the LGSSM global base matrices for 1024 epochs, and then include the GRU network's parameters for the remaining epochs. Our encoder and decoder architecture is based on [7]. In Figure 1, we detail the network architecture used and provide a probabilistic graphical model of the generative process of our combined model.



(a)



(b)



(c)

Fig. 1: Network Architectures. (a) The VAE's encoder that takes in a grayscale image of size 64×64. The blue and red layers output the mean and log of the diagonal covariance of the VAE respectively. (b) The VAE's decoder takes in a measurement a and outputs a grayscale image of size 64×64. (c) The generative graphical model of the complete model with the VAE and LGSSM, figure based on [1]. Diamond nodes represent deterministic variables and shaded nodes represent observed variables. The diamond nodes are the hidden states generated by the GRU network. The GRU network consists of a single layer with 128 hidden units, followed by a fully connected layer with a Softmax non-linearity, which outputs a weighed combinations of base linear matrices as done in [2] and [1].

II. MPC OPTIMIZATION WITH ACTION OR CONTROL REPEATS

The dynamics defined by Equation 6 are learned based on data collected with a control or action repeat n_r [8], where the same control or action is sent for n_r steps when we collect our training trajectories. We use this method in order to better explore the state space through a random Gaussian exploration policy and to maintain smoother controls or actions, without sacrificing the number of training-image-and-control-input pairs collected within a fixed amount of time. To enforce the control repeat in the MPC optimization, we reformulate Equation 12 as

$$
\begin{aligned}
\tilde{\mathbf{u}}^* = \arg\min_{\tilde{\mathbf{u}}} \sum_{k=1}^{T/n_r} &(\mathbf{z}_{n_r k} - \mathbf{z}^g_{T_g} \mathbf{Q}_{\text{mpc}} (\mathbf{z}_{n_r k} - \mathbf{z}^g_{T_g}) \\
&+ \mathbf{u}^T_{n_r k - n_r} \mathbf{R}_{\text{mpc}} \mathbf{u}_{n_r k - n_r}, \\
\text{s.t.} \quad &\mathbf{z}_{n_r(k+1)} = \tilde{\mathbf{A}}_{n_r k} \mathbf{z}_{n_r k} + \tilde{\mathbf{B}}_{n_r k} \mathbf{u}_{n_r k}, \\
&\mathbf{z}_0 = \mathbf{z}^i_{T_i},
\end{aligned} \tag{1}
$$

where $\tilde{\mathbf{u}} = (\mathbf{u}_0, \mathbf{u}_{n_r}, ..., \mathbf{u}_{T-n_r})$ and T is some multiple of n_r. As an example, if $n_r = 3$, $\tilde{\mathbf{A}}_{3k}$ and $\tilde{\mathbf{B}}_{3k}$ are defined recursively by

$$
\begin{aligned}
\mathbf{z}_{3(k+1)} &= \mathbf{A}_{3k+2}(\mathbf{A}_{3k+1}(\mathbf{A}_{3k}\mathbf{z}_{3k} + \mathbf{B}_{3k}\mathbf{u}_{3k}) + \mathbf{B}_{3k+1}\mathbf{u}_{3k}) \\
&\quad + \mathbf{B}_{3k+2}\mathbf{u}_{3k} \\
&= \underbrace{(\mathbf{A}_{3k+2}\mathbf{A}_{3k+1}\mathbf{A}_{3k})}_{\tilde{\mathbf{A}}_{3k}} \mathbf{z}_{3k} + \\
&\quad \underbrace{(\mathbf{A}_{3k+2}\mathbf{A}_{3k+1}\mathbf{B}_{3k} + \mathbf{A}_{3k+2}\mathbf{B}_{3k+1} + \mathbf{B}_{3k+2})}_{\tilde{\mathbf{B}}_{3k}} \mathbf{u}_{3k}
\end{aligned} \tag{2}
$$

REFERENCES

[1] Marco Fraccaro, Simon Kamronn, Ulrich Paquet, and Ole Winther. A disentangled recognition and nonlinear dynamics model for unsupervised learning. In *Adv. Neural Information Processing Systems*, volume 30, pages 3601–3610. 2017.

[2] Maximilian Karl, Maximilian Sölch, Justin Bayer, and Patrick van der Smagt. Deep variational Bayes filters: Unsupervised learning of state space models from raw data. In *5th Int. Conf. Learning Representations*, Toulon, France, Apr. 2017.

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

[4] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proc. 13th Int. Conf. Artificial Intelligence and Statistics*, pages 249–256, 2010.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[6] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[7] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, pages 2450–2462, 2018.

[8] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.