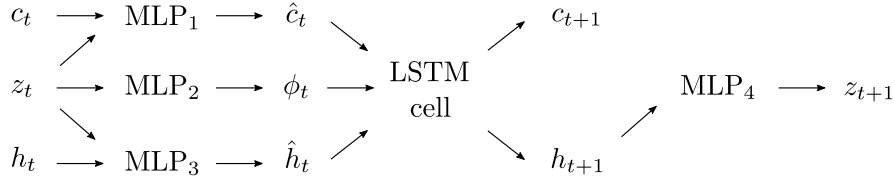


## Architecture details of DSA (improved, better stability)

The dynamics network for  $p(z_{t+1}|z_t)$ :



MLP<sub>1</sub>:  $\text{concat}(z_t, c_t) \rightarrow \hat{c}_t$ , one-layer network, tanh activation.

MLP<sub>2</sub>:  $z_t \rightarrow \phi_t$ , one-layer network, tanh activation.

MLP<sub>3</sub>:  $\text{concat}(z_t, h_t) \rightarrow \hat{h}_t$ , one-layer network, tanh activation.

LSTM cell: apply LSTM equations with  $\phi_t$  as the current input,  $\hat{h}_t$  as the previous hidden state, and  $\hat{c}_t$  as the previous cell state.

MLP<sub>4</sub>:  $h_t \rightarrow \mu, \log \sigma$  of  $p(z_{t+1}|z_t)$ , two-layer network  $[\text{dimH}, \text{dimH}, \text{dimZ} \times 2]$ , tanh activation for the first layer and linear activation for the second layer.

$p(f)$  is simply standard normal.

We need to balance the power of  $p(f)$  and  $p(z_{1:T})$ . This means  $\text{dimZ}$ ,  $\text{dimF}$  and  $\text{dimH}$  need to be chosen carefully. My empirical choice is  $\text{dimZ}=32$ ,  $\text{dimF}=256$  and  $\text{dimH}=256$ .

$p(x_t|z_t, f)$  is defined by a deconvolution neural network with ReLU activation (except the last layer which uses sigmoid). The architecture is like

- $\text{concat}(f, z_t) \rightarrow \Phi_t$  using a two-layer MLP with size  $[\text{dimZ}+\text{dimF}, \text{dimHidden}]$ ,  $4 \times 4 \times \text{nChannel}$ , ReLU activation;
- $\Phi_t \rightarrow x_t$  using a deconv net with filter size 3, shape  $[4 \times 4 \times \text{nChannel}, 8 \times 8 \times \text{nChannel}, 16 \times 16 \times \text{nChannel}, 32 \times 32 \times \text{nChannel}, 64 \times 64 \times \text{nChannel}, 64 \times 64 \times 3]$ , ReLU activation except for the last deconv layer which uses sigmoid.

The dimensions of the deconv net doesn't matter too much for disentanglement, although using a big value for  $\text{dimHidden}$  and  $\text{nChannel}$  would improve the frame quality. I use for example  $\text{dimHidden}=512$  and  $\text{nChannel}=256$ .

I don't think the encoder matters too much in terms of disentanglement, although the image quality can differ. Can simply try the fully factorised one  $q(f, z_{1:T}|x_{1:T}) = q(f|x_{1:T})q(z_t|x_t)$  and share a feature extractor for both  $q(f)$  and  $q(z_t)$ .

### On mixing deterministic and stochastic dynamics

A more rigorous way to write the prior dynamics would be to define  $p(z_{t+1}, h_{t+1}, c_{t+1}|z_t, h_t, c_t)$ , where

$$p(z_{t+1}, h_{t+1}, c_{t+1}|z_t, h_t, c_t) = p(z_{t+1}|h_{t+1}, c_{t+1}, z_t, h_t, c_t)p(h_{t+1}, c_{t+1}|z_t, h_t, c_t),$$

$$p(h_{t+1}, c_{t+1}|z_t, h_t, c_t) = \delta([h_{t+1}, c_{t+1}] = g(z_t, h_t, c_t)),$$

$$p(z_{t+1}|h_{t+1}, c_{t+1}, z_t, h_t, c_t) = p(z_{t+1}|h_{t+1}) = \mathcal{N}(z_{t+1}; \mu(h_{t+1}), \sigma^2(h_{t+1})).$$

Here we use dirac measure for the LSTM states  $h_t$  and  $c_t$ , and we see that this is in fact a mix of deterministic/stochastic dynamics.

For approximate posterior we simply define (for the fully factorised case)

$$q(z_{t+1}, h_{t+1}, c_{t+1}|z_t, h_t, c_t, x_{1:T}) = q(z_{t+1}|x_{t+1})p(h_{t+1}, c_{t+1}|z_t, h_t, c_t).$$

So that it returns the ELBO

$$\mathcal{L} = \sum_{t=1}^T E_{q(f, z_{1:T}|x_{1:T})} \prod_{t=1}^T p(h_t, c_t|z_{t-1}, h_{t-1}, c_{t-1}) \left[ \log \frac{p(f)p(z_t|h_t)p(x_t|f, z_t)}{q(f|x_{1:T})q(z_{t+1}|x_{t+1})} \right],$$

Using LOTUS we can rewrite the lower-bound

$$\mathcal{L} = \sum_{t=1}^T E_{q(f, z_{1:T}|x_{1:T})} \left[ \log \frac{p(f)p(z_t|h_t = \text{prior-net}(z_{t-1}, h_{t-1}, c_{t-1}))p(x_t|f, z_t)}{q(f|x_{1:T})q(z_{t+1}|x_{t+1})} \right].$$