Supplemental Materials:

Online learning in motion modeling for intra-interventional image sequences

Derivation of the ELBO

The conditional probability density function

$$p(\boldsymbol{y} \mid y_0) = \frac{p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \mid y_0)}{p(\boldsymbol{x}, \boldsymbol{z} \mid y_0, \boldsymbol{y})}, \tag{1}$$

is infeasible due to the intractable posterior distribution $p(\boldsymbol{x}, \boldsymbol{z} \mid y_0, \boldsymbol{y})$. Instead, we can approximate the posterior distribution, and identify a lower bound of $p(\boldsymbol{y} \mid y_0)$. In KVAE the posterior distribution is approximated as

$$q(\boldsymbol{x}, \boldsymbol{z} \mid y_0, \boldsymbol{y}) = q_\phi(\boldsymbol{x} \mid y_0, \boldsymbol{y}) p_\gamma(\boldsymbol{z} \mid \boldsymbol{x}), \tag{2}$$

where $q_\phi(\boldsymbol{x} \mid y_0, \boldsymbol{y}) = \prod_{t=1}^{T} q_\phi(x_t \mid y_0, y_t)$ is parameterized using the inference network, i.e.

$$q_\phi(x_t \mid y_0, y_t) = \mathcal{N}(x_t \mid \mu_t^{\text{enc}}, \Sigma_t^{\text{enc}}). \tag{3}$$

If we rewrite the true posterior distribution

$$p(\boldsymbol{x}, \boldsymbol{z} \mid y_0, \boldsymbol{y}) = \frac{p(y_0, \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{z})}{p(y_0, \boldsymbol{y})}, \tag{4}$$

and derive the full distribution model

$$p(y_0, \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{z}) = p(y_0) p_\theta(\boldsymbol{y} \mid y_0, x) p_\gamma(\boldsymbol{x}, \boldsymbol{z}), \tag{5}$$

the true posterior distribution is equivalent to

$$p(\boldsymbol{x}, \boldsymbol{z} \mid y_0, \boldsymbol{y}) = \frac{p(y_0) p_\theta(\boldsymbol{y} \mid y_0, \boldsymbol{x}) p_\gamma(\boldsymbol{x}, \boldsymbol{z})}{p(y_0, \boldsymbol{y})} \tag{6}$$

$$= \frac{p_\theta(\boldsymbol{y} \mid y_0, \boldsymbol{x}) p_\gamma(\boldsymbol{x}, \boldsymbol{z})}{p(\boldsymbol{y} \mid y_0)}. \tag{7}$$

Next, from the KL divergence between the true posterior distribution and our approximate posterior distribution

$$\mathrm{D_{KL}}(q(\boldsymbol{x}, \boldsymbol{z} \mid y_0, \boldsymbol{y}) || p(\boldsymbol{x}, \boldsymbol{z} \mid y_0, \boldsymbol{y})) \geq 0, \tag{8}$$

we have that

$$\mathrm{D_{KL}}(q||p) = \mathbb{E}_{q(\boldsymbol{x}, \boldsymbol{z} \mid y_0, \boldsymbol{y})} \left[\log \frac{q(\boldsymbol{x}, \boldsymbol{z} \mid y_0, \boldsymbol{y})}{p(\boldsymbol{x}, \boldsymbol{z} \mid y_0, \boldsymbol{y})} \right] \tag{9}$$

$$= \mathbb{E}_{q(\boldsymbol{x}, \boldsymbol{z} \mid y_0, \boldsymbol{y})} \left[\log \frac{q_\phi(\boldsymbol{x} \mid y_0, \boldsymbol{y}) p_\gamma(\boldsymbol{z} \mid \boldsymbol{x}) p(\boldsymbol{y} \mid y_0)}{p_\theta(\boldsymbol{y} \mid y_0, \boldsymbol{x}) p_\gamma(\boldsymbol{x}, \boldsymbol{z})} \right] \tag{10}$$

$$= \log p(\boldsymbol{y} \mid y_0) - \mathbb{E}_{q(\boldsymbol{x}, \boldsymbol{z} \mid y_0, \boldsymbol{y})} \left[\log \frac{p_\theta(\boldsymbol{y} \mid y_0, \boldsymbol{x}) p_\gamma(\boldsymbol{x}, \boldsymbol{z})}{q_\phi(\boldsymbol{x} \mid y_0, \boldsymbol{y}) p_\gamma(\boldsymbol{z} \mid \boldsymbol{x})} \right] \geq 0 \tag{11}$$

Finally, by moving the expectation to the right-hand side of the inequality, a tractable lower bound of the likelihood is identified

$$\log p(\boldsymbol{y} \mid y_0) \geq \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{z}|y_0,\boldsymbol{y})}\left[\log \frac{p_\theta(\boldsymbol{y} \mid y_0, \boldsymbol{x}) p_\gamma(\boldsymbol{x}, \boldsymbol{z})}{q_\phi(\boldsymbol{x} \mid y_0, \boldsymbol{y}) p_\gamma(\boldsymbol{z} \mid \boldsymbol{x})}\right]. \tag{12}$$

Model architecture

Encoder (129k & 84k parameters): The inference network and the spatial feature extraction share a similar network architecture. We downsample the data using a stack of convolutional layers, where we extract spatial features at each resolution. The network downsamples the data four times using CNNs with filters $[32, 32, 32, 16]$ and then flattens and feeds the features into a dense network. We approximate the posterior distribution by estimating the mean and covariance of x_t.

Decoder (129k parameters): For the generative network, we use attention gates to focus the temporal changes on the spatial features of the reference image at each resolution, followed by an upsampling CNN. The upsampling uses the same number of resolution layers and filters per level as the downsampling. At the output level, we apply a Gaussian filter (with $\sigma_G = 2$ in the ACDC model and $\sigma_G = 4$ in the EchoNet-Dynamic model) after the last convolutional layer. To enforce diffeomorphic estimates of φ_t, we consider the output as the stationary velocity field v_t and compute the transformation numerically using four scaling and squaring layers.

LG-SSM (976 parameters): We design the LG-SSM using eight dimensions for x_t ($p = 8$) and 16 for the state-variable z_t ($q = 16$). We estimate the full matrices A, C, the initial mean μ_0, and the lower triangular matrices of the covariances R, Q, Σ_0.

Training procedure: For training purposes, we transform the reference image y_0 using the estimated spatial transformation to compute the likelihood $p_\theta(y_t \mid y_0, \varphi_t)$. For the ACDC experiment, we use a local cross-correlation distribution as likelihood and a Gaussian distribution in the EchoNet-dynamic experiment. We optimize the network using Adam optimizer with a learning rate 5×10^{-4} in both the offline and online scenarios. During offline training, we used a batch size of 4 and trained the ACDC model for 500 epochs and the EchoNet-Dynamic model for 50 epochs.