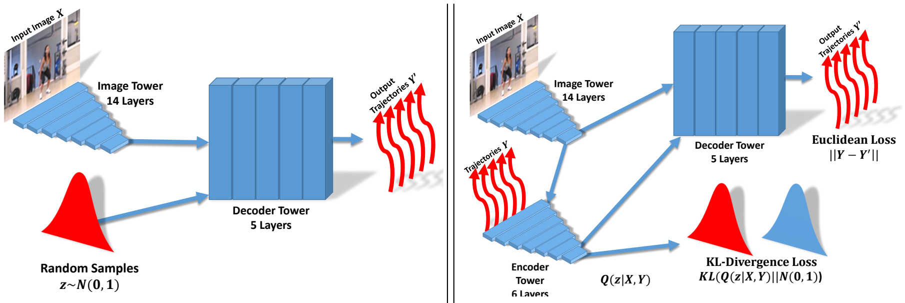

(a) Testing Architecture

(b) Training Architecture

Fig. 2. Overview of the architecture. During training, the inputs to the network include both the image and the ground truth trajectories. A variational autoencoder encodes the joint image and trajectory space, while the decoder produces trajectories depending both on the image information as well as output from the encoder. During test time, the only inputs to the decoder are the image and latent variables sampled from a normal distribution.

$$Y = \mu(X, z) + \epsilon \tag{1}$$

where $z \sim \mathcal{N}(0,1)$, $\epsilon \sim \mathcal{N}(0,1)$ are both white Gaussian noise. We assume μ is implemented as a neural network.

Given a training example (X_i, Y_i), it is difficult to directly infer $P(Y_i|X_i)$ without sampling a large number of z values. Hence, the variational "autoencoder" framework first samples z from some distribution different from $\mathcal{N}(0,1)$ (specifically, a distribution of z values which are likely to give rise to Y_i given X_i), and uses that sample to approximate $P(Y|X)$ in the following way. Say that z is sampled from an arbitrary distribution $z \sim Q$ with p.d.f. $Q(z)$. By Bayes rule, we have:

$$E_{z\sim Q}\left[\log P(Y_i|z, X_i)\right] = E_{z\sim Q}\left[\log P(z|Y_i, X_i) - \log P(z|X_i) + \log P(Y_i|X_i)\right] \tag{2}$$

Rearranging the terms and subtracting $E_{z\sim Q} \log Q(z)$ from both sides:

$$\begin{aligned} \log P(Y_i|X_i) - E_{z\sim Q}\left[\log Q(z) - \log P(z|X_i, Y_i)\right] = \\ E_{z\sim Q}\left[\log P(Y_i|z, X_i) + \log P(z|X_i) - \log Q(z)\right] \end{aligned} \tag{3}$$

Note that X_i and Y_i are fixed, and Q is an arbitrary distribution. Hence, during training, it makes sense to choose a Q which will make $E_{z\sim Q}[\log Q(z) - \log P(z|X_i, Y_i)]$ (a $\mathcal{KL}$-divergence) small, such that the right hand side is a close approximation to $\log P(Y_i|X_i)$. Specifically, we set $Q = \mathcal{N}(\mu'(X_i, Y_i), \sigma'(X_i, Y_i))$ for functions μ' and σ', which are also implemented as neural networks, and which are trained alongside μ. We denote this p.d.f. as $Q(z|X_i, Y_i)$. We can rewrite some of the above expectations as $\mathcal{KL}$-divergences to obtain the standard variational equality:

$$\begin{aligned} \log P(Y_i|X_i) - \mathcal{KL}\left[Q(z|X_i, Y_i) \| P(z|X_i, Y_i)\right] = \\ E_{z\sim Q}\left[\log P(Y_i|z, X_i)\right] - \mathcal{KL}\left[Q(z|X_i, Y_i) \| P(z|X_i)\right] \end{aligned} \tag{4}$$

We compute the expected gradient with respect to only the right hand side