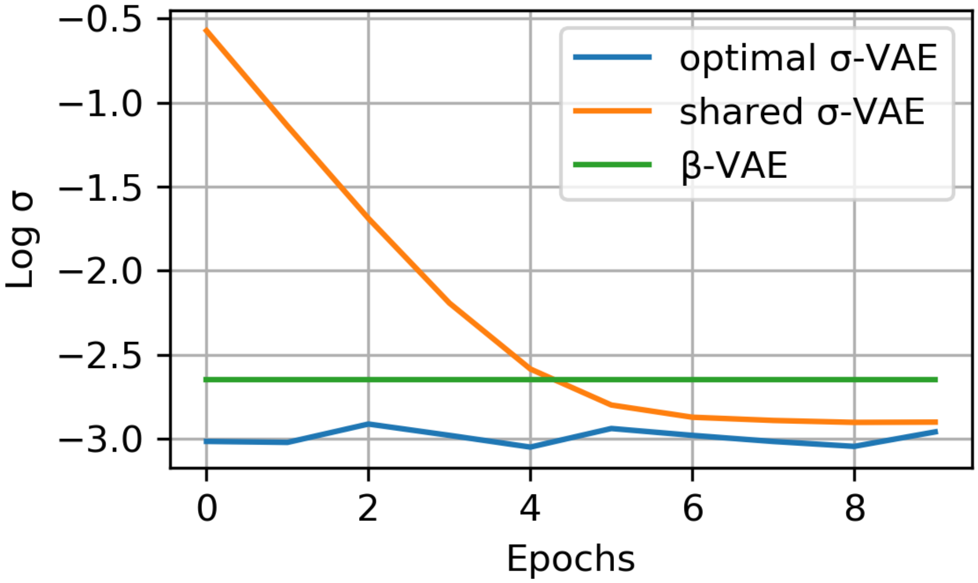

Figure 5: Variance convergence speed on SVHN. x-axis: epochs, y-axis: $\ln \sigma$. We see that the shared $\sigma$-VAE which optimizes the variance with gradient descent has an initial period of convergence when the variance converges to the region of the optimal value. In contrast, $\sigma$-VAE with analytical (optimal) variance quickly learns a good estimate of the variance, which leads to better performance. The unit variance Gaussian $\beta$-VAE can be intepreted as having a constant variance determined by $\beta$, shown here. Since the variance doesn't change throughout training, it achieves suboptimal performance.

## A Additional experimental results

In this section, we provide more qualitative results as well as a graph showing the convergence properties of the variance for different models.

## B Experimental details

For the image VAE models, the encoder has 3 convolutional layers followed by a fully connected layer, while the decoder has a fully connected layer followed by 3 convolutional layers. The $\beta$ was tuned from 100 to 0.0001 for $\beta$-VAE. The number of channels in the convolutional layers starts with 32 and increases 2 times in every layer, except on the CIFAR data, where it starts with 128. The dimension of the latent variable is 20. Adam [24] with learning rate 1e-3 is used for optimization. Batch size of 128 was used and all models were trained for 10 epochs. Unit Gaussian prior and Gaussian posteriors with diagonal covariance were used. For the SVG models, the original hyperparameters for the SVG-LP model were used. We use the standard train-val-test split for all datasets. All models were trained on a single high-end GPU.

Table 4: ELBO on discretized data. All distributions except categorical have scalar scale parameters. The $\sigma$-VAE performs well on the discretized ELBO metric, performing similarly to a discrete distribution parametrized as a discretized Gaussian or discretized Logistic. Full categorical distribution attains highest likelihood due to having the most statistical power.

| | CIFAR VAE | | |
|---|---|---|---|
| | $-\log \text{pdf} \downarrow$ | $-\log p \downarrow$ | FID $\downarrow$ |
| Categorical VAE | | $<$ **10673** | **137.6** |
| Gaussian VAE | $< 740.5$ | $< 15131$ | 212.7 |
| Gaussian $\sigma$-VAE | $< -896.1$ | $< 11120$ | **136.7** |
| Disc. Gaussian $\sigma$-VAE | | $< 11117$ | **136.9** |
| Disc. Logistic $\sigma$-VAE | | $< 11103$ | **136.7** |

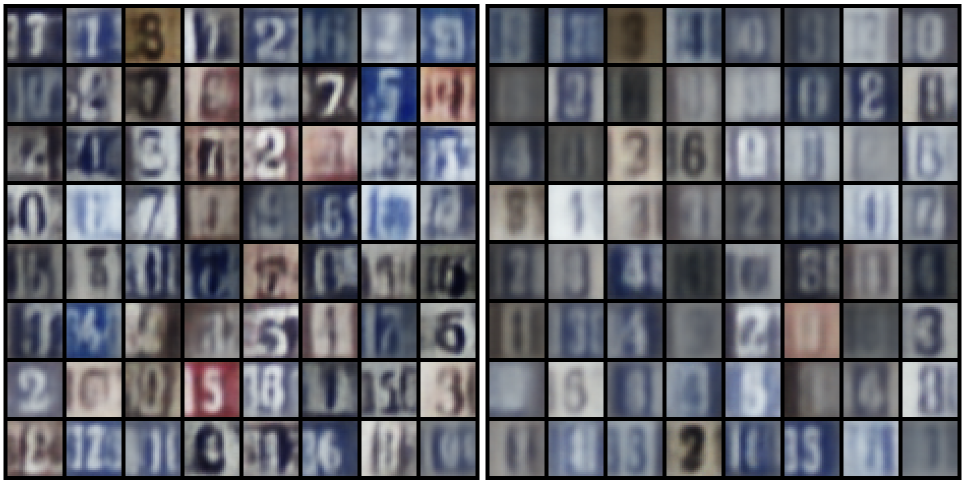

Figure 6: Samples from the $\sigma$-VAE (left) and the Gaussian VAE (right) on the SVHN dataset. The Gaussian VAE produces blurry results with muted colors, while the $\sigma$-VAE is able to produce accurate images of digits.

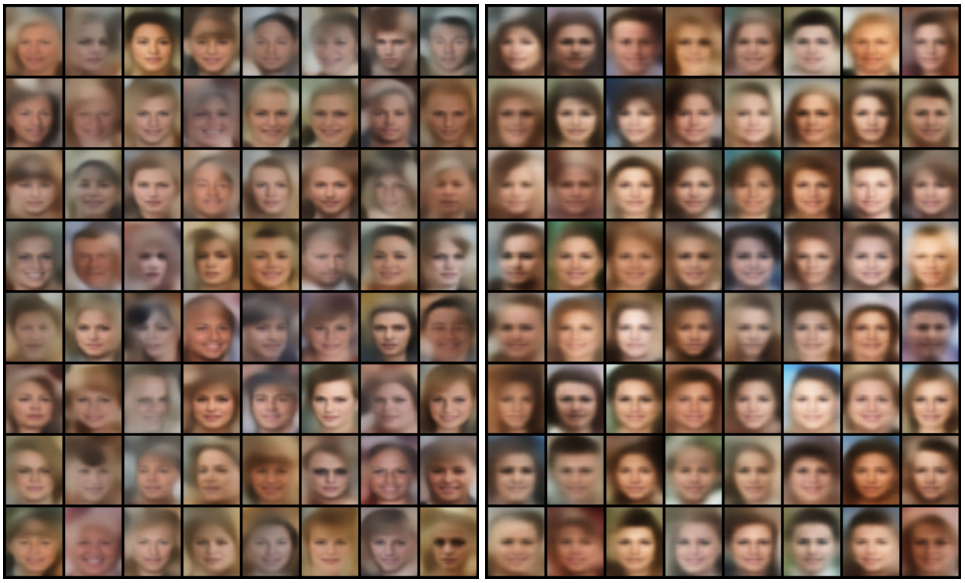

Figure 7: Samples from the $\sigma$-VAE (left) and the Gaussian VAE (right) on the CelebA dataset, images cropped to the face for clarity. The Gaussian VAE produces blurry results with indistinct face features, while the $\sigma$-VAE is able to produce accurate images of faces.

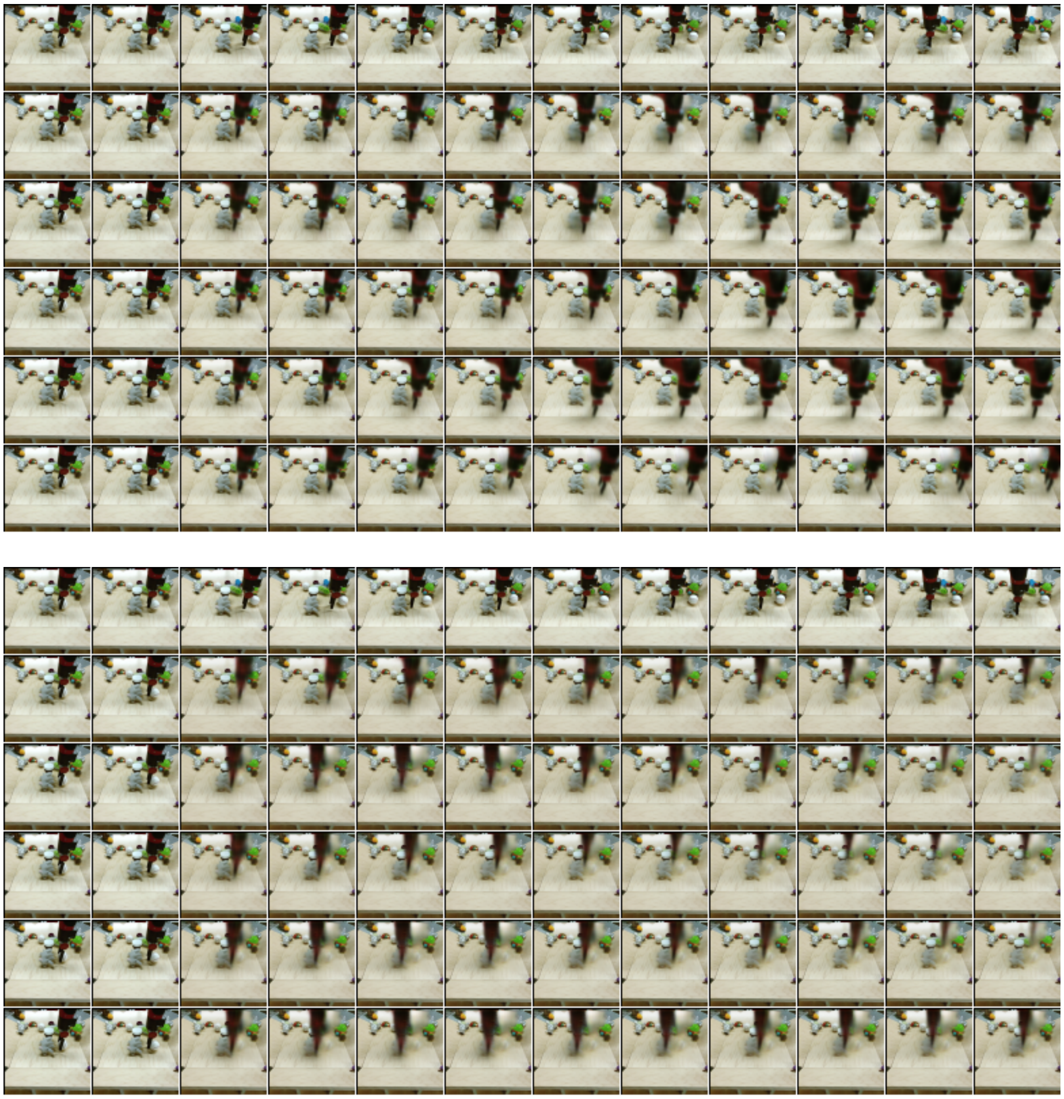

Figure 8: Samples from the $\sigma$-VAE (top) and the Gaussian VAE (bottom) on the BAIR dataset. Sampled sequences conditioned on two initial frames are shown, and the ground truth sequence is shown at the top. The Gaussian VAE produces blurry robot arm texture and the arm often disappears towards the end of the sequence, while the $\sigma$-VAE is able to produce sequences with realistic motion and model the details of the arm texture, such as the gripper.

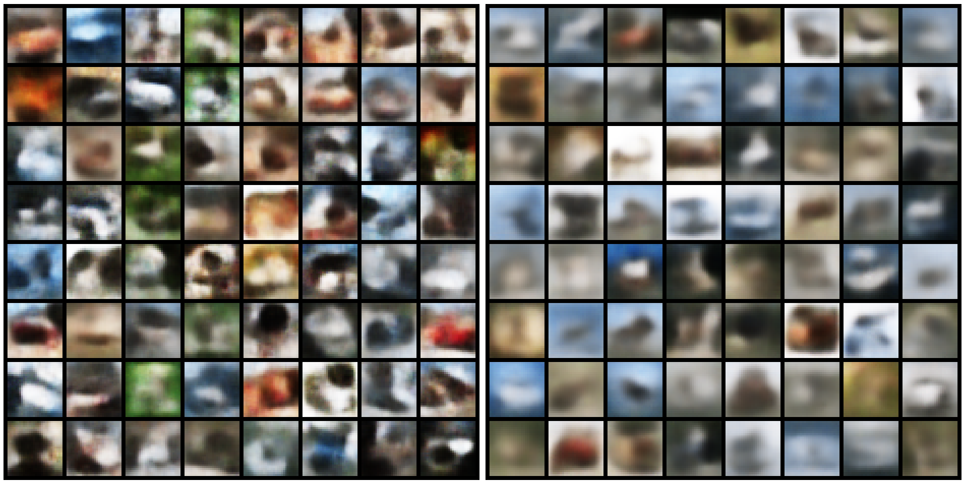

Figure 9: Samples from the $\sigma$-VAE (left) and the Gaussian VAE (right) on the challenging CIFAR dataset. The Gaussian VAE produces blurry results with muted colors, while the $\sigma$-VAE models the distribution of shapes and colors in the CIFAR data more faithfully.

## C Alternative Decoder Choices

We describe the alternative decoders evaluated in Table 2: using the Beta, the bitwise-categorical, and the logistic mixture distributions.

**Beta VAE** Previously described continuous distributions such as Gaussian had positive density on the whole real line, $(-\infty, \infty)$. This might be undesirable since pixel intensity values lie in a finite range, usually scaled to $[0, 1]$. These continuous distributions therefore assign positive densities to impossible intensity values, potentially leading to poor likelihood and invalid samples. We therefore evaluate a continuous distribution that is defined on the $[0, 1]$ range, specifically, the Beta distribution. We parametrize the Beta distribution with the two concentration parameters produced per pixel and channel. We experimented with alternative location-scale parametrizations, however, we found that the parametrization via concentration parameters allows to enforce that the distribution is defined on the inclusive interval $[0, 1]$, by restricting both concentration parameters to be higher than $1$. This is harder to enforce with location-scale parametrizations. In our experiments, we found that the Beta distribution does not outperform the $\sigma$-VAE decoder, as the Gaussian $\sigma$-VAE decoder can ensure that the values outside of the $[0, 1]$ range have small density values by setting the variance to be small. However, we expect the Beta distribution to be useful for enforcing that the model only assigns positive densities to values in a certain range.

**Bitwise-categorical VAE** While the 256-way categorical decoder described in Section 3.2 is very powerful due to the ability to specify any possible intensity distribution, it suffers from high computational and memory requirements. Because 256 values need to be kept for each pixel and channel, simply keeping this distribution in memory for one 3-channel $1024 \times 1024$ image would require 3 GiB of memory, compared to 0.012 GiB for the Gaussian decoder. Therefore, training deep neural networks with this full categorical distribution is impractical for high-resolution images or videos. The bitwise-categorical VAE improves the memory complexity by defining the distribution over 256 values in a more compact way. Specifically, it defines a binary distribution over each bit in the pixel intensity value, requiring 8 values in total, one for each bit. This distribution can be thought of as a classifier that predicts the value of each bit in the image separately. In our implementation of the bitwise-categorical likelihood, we convert the image channels to binary format and use the standard binary cross-entropy loss (which reduces to binary log-likelihood since all bits in the image are deterministically either zero or one). While in our experiments the bitwise-categorical distribution did not outperform other choices, it often performs on par with our proposed method. We expect this distribution to be useful due to its generality as it is able to represent values stored in any digital format by converting them into binary.

**Logistic mixture VAE** For this decoder, we adapt the discretized logistic mixture from Salimans et al. [44]. To define a discrete 256-way distribution, it divides the corresponding continuous distribution into 256 bins, where the probability mass is defined as the integral of the PDF over the corresponding bin. [26] uses the logistic distribution discretized in this manner for the decoder. Salimans et al. [44] suggests to make all bins except the first and the last be of equal size, whereas the first and the last bin include, respectively, the intervals $(-\infty, 0]$ and $[1, \infty)$. Salimans et al. [44] further suggests using a mixture of discretized logistics for improved capacity. Our implementation largely follows the one in Salimans et al. [44], however, we note that the original implementation is not suitable for learning latent variable models, as it generates the channels autoregressively. This will cause the latent variable to lose color information since it can be represented by the autoregressive decoder. We therefore adapt the mixture of discretized logistics to the pure latent variable setup by removing the mean-adjusting coefficients from [44]. In our experiments, the logistic mixture outperformed other discrete distributions.