# Video Generation using multimodal VAE

Anubhav Mittal
anubhavm@iitk.ac.in
Supervised by: Ravindra Yadav, Prof. Vinay P. Namboodiri and Prof. K. Vasudevan

***Abstract*—The discipline of generative modeling has experienced enormous leaps in capabilities in recent years, mostly with likelihood-based methods (Graves, 2013; Kingma and Welling, 2013; van den Oord et al., 2016a) and generative adversarial networks (GANs) (Goodfellow et al., 2014). After giving a brief overview of the methods, we move onto the describe the existing literature on audio and video generation using these methods.**

**We then introduce a new method for video generation, which we base on the multimodal VAE model defined in Wu and Goodman, 2018. We did not achieve respectable results on a simple version of the model, but we found the approach interesting, so we introduce changes to the model by replacing the MSE error with a learned similarity metric (Larsen et al., 2016). We were able to achieve good results on the celebA dataset using this model.**

**Finally, we discuss a class of flow based generative models (Dinh et al., 2014, 2016; Kingma and Dhariwal, 2018) which has gained popularity in recent times. We also suggest how we can use these approaches in video generation.**

## I. Preliminaries: Generative modelling

What does ”generative” mean? ”Generative” describes a class of statistical models that contrasts with discriminative models.

Informally,

1) Generative models can generate new data instances.
2) Discriminative models discriminate between different kinds of data instances.

A generative model could generate new photos of animals that look like real animals, while a discriminative model could tell a dog from a cat. GANs are just one kind of generative model.

More formally, given a set of data instances X and a set of labels Y ,

1) Generative models capture the joint probability p(X, Y), or just p(X) if there are no labels. A generative model includes the distribution of the data itself, and tells you how likely a given example is. For example, models that predict the next word in a sequence are typically generative models because they can assign a probability to a sequence of words.
2) Discriminative models capture the conditional probability $p(Y|X)$. A discriminative model ignores the question of whether a given instance is likely, and just tells you how likely a label is to apply to the instance.

We discuss the most popular of the generative modelling methods, namely GANs and VAEs.

### A. Generative Adversarial Network (GAN)

Generative Adversarial Networks takes up a game-theoretic approach, unlike a conventional neural network. The network learns to generate from a training distribution through a 2-player game. The two entities are Generator and Discriminator. These two adversaries are in constant battle throughout the training process. Since an adversarial learning method is adopted, we need not care about approximating intractable density functions.

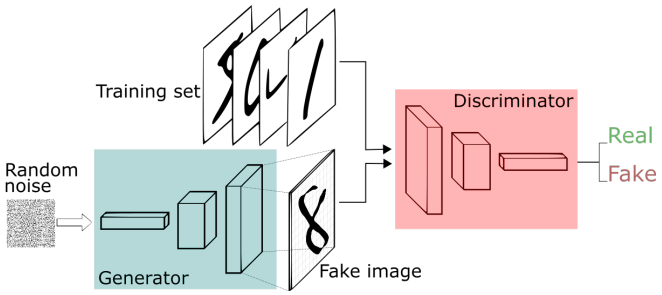


Fig. 1. Generative Adversarial Network (Image courtesy: towardsdata-science.com)

*1) Approach:* As one can identify from their names, a generator is used to generate real-looking images and the discriminator's job is to identify which one is a fake. The entities/adversaries are in constant battle as one(generator) tries to fool the other(discriminator), while the other tries not to be fooled. To generate the best images you will need a very good generator and a discriminator. This is because if your generator is not good enough, it will never be able to fool the discriminator and the model will never converge. If the discriminator is bad, then images which make no sense will also be classified as real and hence your model never trains and in turn you never produces the desired output. The input, random noise can be a Gaussian distribution and values can be sampled from this distribution and fed into the generator network and an image is generated. This generated image is compared with a real image by the discriminator and it tries to identify if the given image is fake or real.

*2) Objective function:* Since a game-theoretic approach is taken, our objective function is represented as a minimax function. The discriminator tries to maximize the objective function, therefore we can perform gradient ascent on the objective function. The generator tries to minimize the objective function, therefore we can perform gradient descent on the objective function. By alternating between gradient ascent and descent, the network can be trained.

$$\min_{\theta_g}\max_{\theta_d}\left[\mathbb{E}_{x\sim p_{data}}\log \underbrace{D_{\theta_d}(x)}_{\substack{\text{Discriminator output}\\ \text{for real data x}}} + \mathbb{E}_{z\sim p(z)}\log(1-\underbrace{D_{\theta_d}(G_{\theta_g}(z))}_{\substack{\text{Discriminator output for}\\ \text{generated fake data G(z)}}})\right]$$

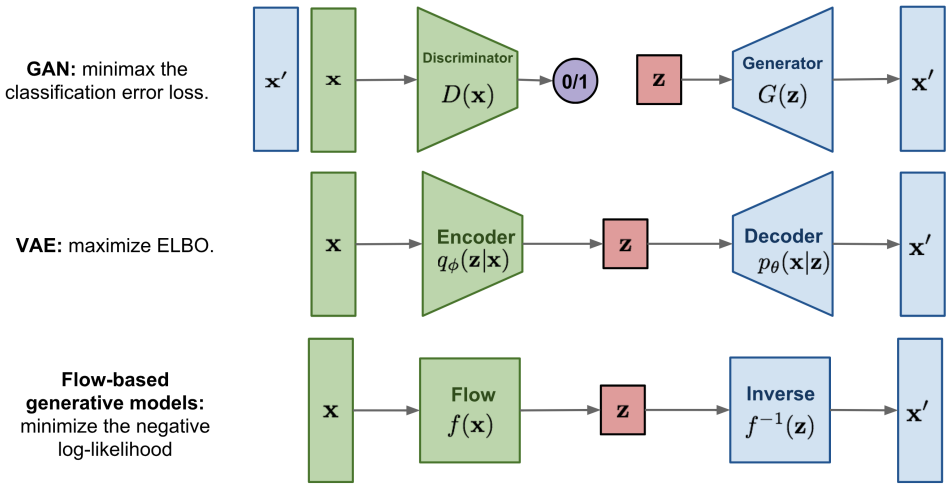


Fig. 2. Three major kinds of generative models. Image courtesy: lilianweng.github.io

For discriminator we simply do the ascent:

$$\max_{\theta_d} \left[ \mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

For generator, if we simply do the descent,

$$\min_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))$$

it is observed that optimizing the generator objective function does not work so well, this is because when the sample is generated is likely to be classified as fake, the model would like to learn from the gradients but the gradients turn out to be relatively flat. This makes it difficult for the model to learn. Therefore, the generator objective function is changed as below.

$$\max_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(D_{\theta_d}(G_{\theta_g}(z)))$$

Instead of minimizing the likelihood of discriminator being correct, we maximize the likelihood of discriminator being wrong. Therefore, we perform gradient ascent on generator according to this objective function.

### B. Variational Autoencoder (VAE)

*1) Autoencoders:* Autoencoders are an unsupervised learning technique in which we leverage neural networks for the task of representation learning. Specifically, we design a neural network architecture such that we impose a bottleneck in the network which forces a compressed knowledge representation of the original input. If the input features were each independent of one another, this compression and subsequent reconstruction would be a very difficult task. However, if some sort of structure exists in the data (ie. correlations between input features), this structure can be learned and consequently leveraged when forcing the input through the network's bottleneck.

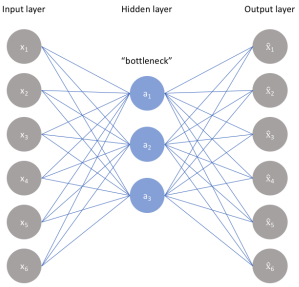


Fig. 3. Autoencoder. Image courtesy: jeremyjordan.me

As visualized above, we can take an unlabeled dataset and frame it as a supervised learning problem tasked with outputting $x'$, a reconstruction of the original input x. This network can be trained by minimizing the reconstruction error, $L(x, x')$, which measures the differences between our original input and the consequent reconstruction.

*2) Auto Encoding Variational Bayes:* A variational autoencoder (VAE) provides a probabilistic manner for describing an observation in latent space. Thus, rather than building an encoder which outputs a single value to describe each latent state attribute, we'll formulate our encoder to describe a probability distribution for each latent attribute.

Instead of encoder outputting an encoding vector of size n, rather, outputting two vectors of size n: a vector of means, $\mu$, and another vector of standard deviations, $\sigma$. We sample the latent variable from this distribution, feed it to the decoder, and backpropagate the loss

$$l_i(\theta, \phi) = -\mathbb{E}_{z \sim q_\theta(z|x_i)}[\log p_\phi(x_i \mid z)] + \mathbb{KL}(q_\theta(z \mid x_i) \mid\mid p(z))$$

through the network, training the encoder and decoder simultaneously. The first term is the reconstruction loss, while the second term ensures that the latent variable distribution

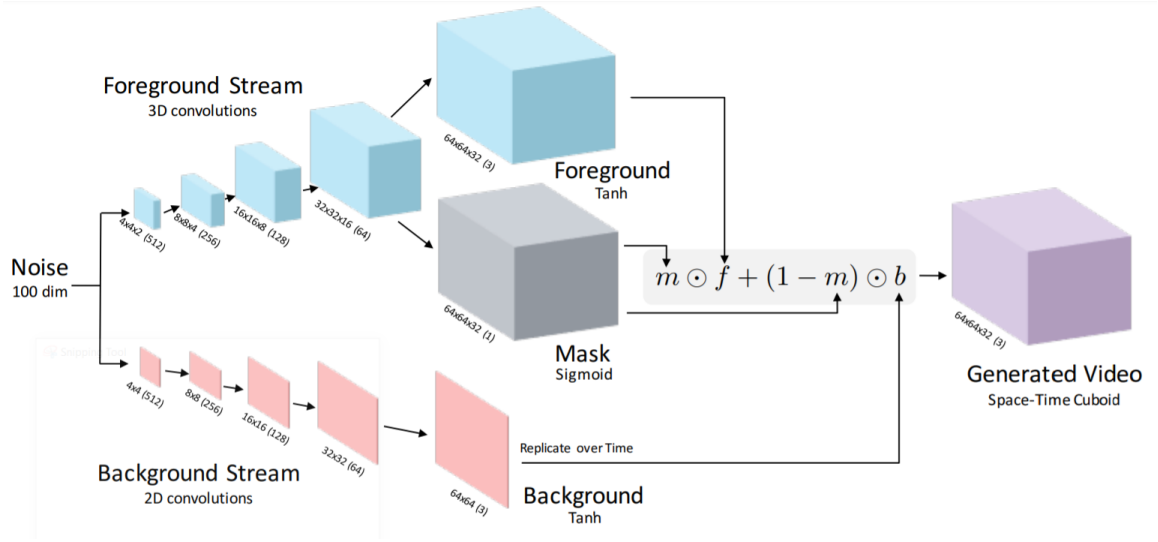


Fig. 4. Video generator network. Image courtesy: Vondrick et al., 2016

stay close to their prior distribution (generally the normal distribution).

## II. Related works

### A. Video generation using GANs

We discuss the approach taken by Vondrick et al., 2016. They use a GAN to generate videos. They present a single and a two stream architecture, where the two stream architecture consist of two separate generator networks, one which generates the foreground representing the objects which move in time, and the other generates the background which remains static in time. We only discuss the two-stream architecture.

*1) Generator Network:* They use a two-stream architecture that enforces a static background and moving foreground. The generator is governed by the combination:

$$G_2(z) = f(z) \odot m(z) + b(z) \odot (1 - m(z))$$

The $0 \geq m(z) \geq 1$ can be viewed as a spatio-temporal mask that selects either the foreground f(z) model or the background model b(z) for each pixel location and timestep. To enforce a background model in the generations, b(z) produces a spatial image that is replicated over time, while f(z) produces a spatio-temporal cuboid masked by m(z). By summing the foreground model with the background model, they obtain the final generation.

They use fractionally strided convolutional networks for upsampling from $z$ to $m(z)$, $f(z)$, and $b(z)$. For $f(z)$, they combine spatio-temporal convolutions with fractionally strided convolutions to generate video. Three dimensional convolutions provide spatial and temporal invariance, while fractionally strided convolutions can upsample efficiently in a deep network, allowing z to be low-dimensional. For $b(z)$ they only use fractionally strided convolution for upsampling. To create the mask m(z), they use a network that shares weights with f(z) except the last layer, which has only one output channel. They use a sigmoid activation function for the mask.

*2) Discriminator network:* The discriminator needs to be able to solve two problems: firstly, it must be able to classify realistic scenes from synthetically generated scenes, and secondly, it must be able to recognize realistic motion between frames. They design the discriminator to be able to solve both of these tasks with the same model. They use a five-layer spatio-temporal convolutional network with kernels 4 4 4 so that the hidden layers can learn both visual models and motion models. The architecture is the reverse of the foreground stream in the generator, replacing fractionally strided convolutions with strided convolutions (to down-sample instead of up-sample), and replacing the last layer to output a binary classification (real or not).

*3) Learning:* They train the generator and discriminator with stochastic gradient descent. Following the usual GAN training process, they alternate between maximizing the loss w.r.t. $w_D$ and minimizing the loss w.r.t. $w_G$ until a fixed number of iterations.

The final results produced by the model, although very weak in themselves, are very encouraging and were one of the first attempts at video generation using generative models.

### B. Video generation using VAEs

We discuss the approach taken by Denton and Fergus, 2018. This appeared at ICLR 2018 along with a broadly similar approach by Babaeizadeh et al. . However, as the former's approach is more intuitive and achieves better results, we only discuss the later's approach in brief at the end.

The model has two distinct components: (i) a prediction model $p_\theta$ that generates the next frame $x'_t$, based on previous ones in the sequence $x_{1:t-1}$ and a latent variable $z_t$ and (ii) a prior distribution $p(z)$ from which $z_t$ is sampled at at each time step . The prior distribution can be fixed (SVG-FP) or learned (SVG-LP). Intuitively, the latent variable $z_t$ carries all the stochastic information about the next frame that the deterministic prediction model cannot capture. After conditioning on a short series of real frames, the model can

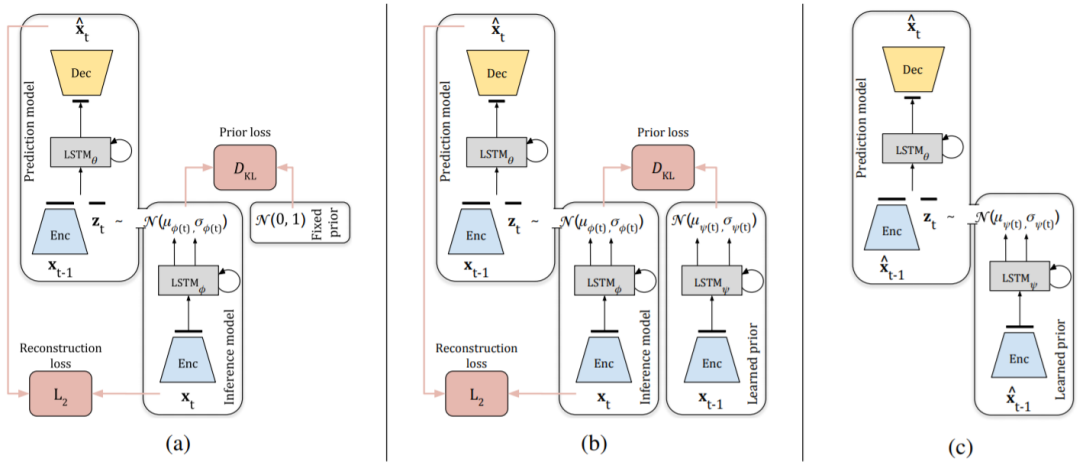


Fig. 5. The video generation model by Denton and Fergus. (a) Training with a fixed prior (SVG-FP); (b) Training with learned prior (SVG-LP); (c) Generation with the learned prior model. The red boxes show the loss functions used during training. Image courtesy: Denton and Fergus, 2018

generate multiple frames into the future by passing generated frames back into the input of the prediction model and, in the case of the SVG-LP model, the prior also.

The recurrent frame predictor $p_\theta(x'_t|x_{1:t-1}, z_{1:t})$ is specified by a fixed-variance conditional Gaussian distribution $\mathcal{N}(\mu_\theta(x_{1:t-1}, z_{1:t}), \sigma)$. In practice, we set $x'_t = \mu_\theta(x_{1:t-1}, z_{1:t})$, i.e. the mean of the distribution, rather than sampling. Note that at time step t the frame predictor only receives $x_{t-1}$ and $z_t$ as input. The dependencies on all previous $x_{1:t-2}$ and $z_{1:t-1}$ stem from the recurrent nature of the model, which is modelled by an LSTM. Since the true distribution over latent variables $z_t$ is intractable we rely on a time-dependent inference network $q_\phi(z_t|x_{1:t})$ that approximates it with a conditional Gaussian distribution $\mathcal{N}(\mu_\phi(x_{1:t}), \sigma_\phi(x_{1:t}))$. The model is trained by optimizing the variational lower bound:

$$\mathcal{L}_{\theta,\phi}(\mathbf{x}_{1:T}) = \sum_{t=1}^{T} \big[\mathbb{E}_{q_\phi(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})} \log p_\theta(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t}) - \beta D_{KL}(q_\phi(\mathbf{z}_t|\mathbf{x}_{1:t})||p(\mathbf{z}))\big]$$

*1) Fixed Prior:* The simplest choice for $p(z_t)$ is a fixed Gaussian $\mathcal{N}(0, I)$, as is typically used in variational auto encoder models. This is called the SVG-FP model, as shown in in Fig. 5a. A drawback is that samples at each time step will be drawn randomly, thus ignore temporal dependencies present between frames.

*2) Learned prior:* A more sophisticated approach is to learn a prior that varies across time, being a function of all past frames up to but not including the frame being predicted $p_\psi(z_t|x_{1:t-1})$. Specifically, at time t a prior network observes frames $x_{1:t-1}$ and output the parameters of a conditional Gaussian distribution $\mathcal{N}(\mu_\psi(x_{1:t-1}), \sigma_\psi(x_{1:t-1}))$. The prior network is trained jointly with the rest of the model by maximizing:

$$\mathcal{L}_{\theta,\phi,\psi}(\mathbf{x}_{1:T}) = \sum_{t=1}^{T} \big[\mathbb{E}_{q_\phi(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})} \log p_\theta(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t}) - \beta D_{KL}(q_\phi(\mathbf{z}_t|\mathbf{x}_{1:t})||p_\psi(\mathbf{z}_t|\mathbf{x}_{1:t-1}))\big]$$

This model is called SVG-LP and and its training procedure is shown in Fig. 5b.

At test time, a frame at time $t$ is generated by first sampling $z_t$ from the prior. In SVG-FP draw $z_t$ from $\mathcal{N}(0, I)$ and in SVG-LP we draw $z_t$ from $p_\psi(z_t|x_{1:t-1})$. Then, a frame is generated by $x'_t = \mu_\theta(x_{1:t-1}, z_{1:t})$. After conditioning on a short series of real frames, the model begins to pass generated frames $x'_t$ back into the input of the prediction model and, in the case of the SVG-LP model, the prior. The sampling procedure for SVG-LP is illustrated in Fig. 5c.

The final results by the model are very good both qualitatively and quantitatively. We use this approach as inspiration in designing our model.

### C. Other works on generation

We also went through other related works such as Video to audio generation in Ephrat et al., 2017 and Zhou et al., 2018. These use Recurrent CNNs and train on different kind of videos (Human face centered speech in the former and random videos from the wild in the later). Afterwards, they try to generate audio for silent videos of the same/similar dataset. They show encouraging results and may be incorporated in some form in future video generation models, although we could not design suitable models for now.

## III. Work done

We will first describe a multimodal VAE model that we based our approach on, and then describe the approach. As the model did not achieve good qualitative results for cases simpler than video generation, we change the reconstruction loss function from MSE to a learned similarity metric. Unfortunately, this model also failed to achieve respectable results.

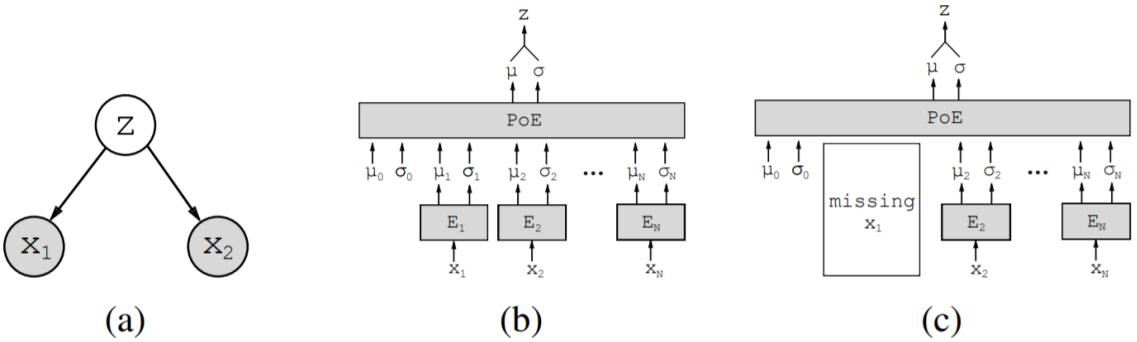


Fig. 6. The model by Wu and Goodman. : (a) Graphical model of the MVAE. Gray circles represent observed variables. (b) MVAE architecture with N modalities. $E_i$ represents the i-th inference network; $\mu_i$ and $\sigma_i$ represent the i-th variational parameters; $\mu_0$ and $\sigma_0$ represent the prior parameters. The product-of-experts (PoE) combines all variational parameters in a principled and efficient manner. (c) If a modality is missing during training, we drop the respective inference network. Thus, the parameters of $E_1, ..., E_N$ are shared across different combinations of missing inputs.. Image courtesy: Wu and Goodman, 2018

### A. Multimodal generative model

We discuss the model specified in Wu and Goodman, 2018, as we are going to use it in our model. They present a model which first trains a VAE on multiple modalities, and capture the relationship between them. Then, using the generative structure of VAE, they show that we can generate missing modalities in presence of other modalities.

In the multimodal setting we assume the N modalities, $x_1, ..., x_N$ , are conditionally independent given the common latent variable, $z$ (See Fig. 6a). That is we assume a generative model of the form $p_\theta(x_1, x_2, ..., x_N, z) = p(z)p_\theta(x_1|z)p_\theta(x_2|z)...p_\theta(x_N|z)$. With this factorization, we can ignore unobserved modalities when evaluating the marginal likelihood. If we write a data point as the collection of modalities present, that is $X = \{x_i | i^{th}$ modality present$\}$, then the ELBO becomes:

$$\text{ELBO}(X) \triangleq \mathbb{E}_{q_\phi(z|X)}[\sum_{x_i \in X} \lambda_i \log p_\theta(x_i|z)] - \beta \, \text{KL}[q_\phi(z|X), p(z)].$$

They approximate the joint posterior as a product of experts of individual posteriors computed by the encoder network of VAE and the prior for latent variables. However, this will not result in the model learning the interdependencies between the modalities. We also cannot compute the $2^N$ subsets independently and then add up all the losses, as it will be computationally intractable. So they propose the loss function for an iteration to be the sum of the joint ELBO, individual ELBOs and k ELBO terms using k randomly chosen subsets, $X_k$. For each minibatch, we thus evaluate a random subset of the 2 N ELBO terms. In expectation, we will be approximating the full objective. The sub-sampled objective can be written as

$$\text{ELBO}(x_1, ..., x_N) + \sum_{i=1}^{N} \text{ELBO}(x_i) + \sum_{j=1}^{k} \text{ELBO}(X_j)$$

This is the final loss function, which they propagate through the network and train the sub-inference networks. The experiments show that they are able to learn the dependencies between different modes at training time, and come test time, they are able to predict missing modalities with respectable accuracy.

### B. Method-I

*1) Approach:* We use the multimodal model described above for video generation. The model is shown in Fig 7. Specifically, there are 2 modalities - the previous video frame $x_{t-1}$ and the current video frame $x_t$. During training, these are passed into encoders followed by a LSTM, which gives us $\mu_a, \sigma_a$ and $\mu_b, \sigma_b$ respectively. Using the product of experts approach defined above, we compute loss according to the total ELBO:

$$ELBO(x_a, x_b) + ELBO(x_a) + ELBO(x_b)$$

where $x_a, x_b$ correspond to $x_t, x_{t-1}$ respectively. At test time, following an initial conditioning, the model should accept $x_{t-1}$ as input, and produce both $x_{t-1}$ and $x_t$ as output.

*2) Experiments:* We trained the model on the moving MNIST dataset. The model parameters were similar those used in the Wu paper. However, the results were poor.

To check if the model could learn simpler correlations, we ran it on the celebA dataset, with the first modality as the images and the second modality as feature vectors. During test time, this model outputed images which resembled human faces, but were not near to the kind of accuracy required for video generation.

### C. Learned similarity metric

The major reason for blurry images in the VAE reconstruction is attributed to the MSE loss between the original image and the VAE output. Element-wise reconstruction errors are not adequate for images and other signals with invariances. So in the work by Larsen et al., 2016, they propose to exploit an appealing property of GAN, that its discriminator network implicitly has to learn a rich similarity metric for images, so as to discriminate them from "non-images". The end result will be a method that combines the advantage of GAN as a high

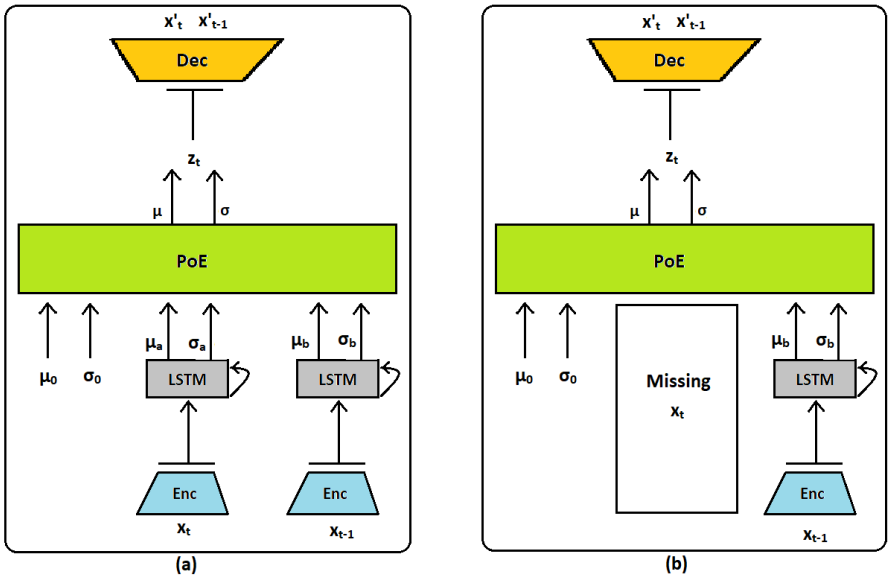


Fig. 7. The first proposed model. (a) During training, both $x_t$ and $x_{t-1}$ are available. The model learns to generate both $x'_t$ and $x'_{t-1}$ and the dependency between the two. (b) During test time, only $x_{t-1}$ is present. As the model has learned the dependency between the current and the next frame, it should be able to generate the next frame.

quality generative model and VAE as a method that produces an encoder of data into the latent space z.

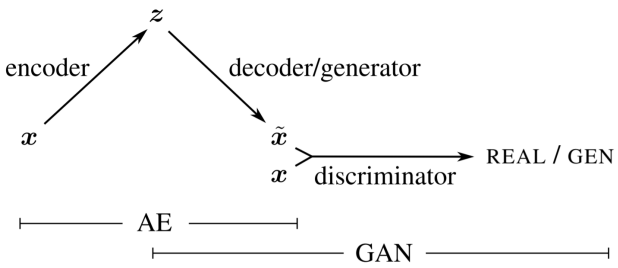


Fig. 8. Overview of the network in Larsen et al. 2016. Image courtesy: their original paper.

Specifically, they propose replacing the VAE reconstruction (expected log likelihood) error term with a reconstruction error expressed in the GAN discriminator. To achieve this, let $Dis_l(x)$ denote the hidden representation of the $l^{th}$ layer of the discriminator. They introduce a Gaussian observation model for $Dis_l(x)$ with mean $Dis_l(x')$ and identity covariance:

$$p(Dis_l(x)|z) = \mathcal{N}(Dis_l(x)|Dis_l(x'), I)$$

where $x'$ is a sample from the decoder of VAE with $x$ as input. Hence, the reconstruction loss of VAE is replaced by:

$$\mathcal{L}_{llike}^{Dis_l} = E_{q(z|x)}[logp(Dis_l(x)|z)]$$

The total loss is given by the triple criterion:

$$\mathcal{L} = \mathcal{L}_{prior} + \mathcal{L}_{llike}^{Dis_l} + \mathcal{L}_{GAN}$$

### D. Method-II

*1) Approach:* As stated above, because of the problems of the normal reconstruction MSE error, we add a GAN discriminator at the output of the decoder/generator. Similarly, we change the loss function to the one described above, so that the new ELBO is :

$$\begin{aligned} ELBO(X) =& E_{q_\phi(z|X)}[\sum_{x_i \in X} \log p(Dis_l(x_i)|z)] + \\ & \sum_{x_i \in X} [E_{x_i \sim p_{data}} \log D_i(x) + \\ & E_{q_\phi(z|X)} \log(1 - D_i(Dec(z)_i))] \\ & - \beta KL[q_\phi(z|X), p(z)] \end{aligned}$$

The first term is the modified reconstruction error, the second and third term are the GAN loss and the fourth term is the KL divergence between the posterior and its prior. The sub-sampled objective can be the same as before (in expectation):

$$\text{ELBO}(x_1, ..., x_N) + \sum_{i=1}^{N} \text{ELBO}(x_i) + \sum_{j=1}^{k} \text{ELBO}(X_j)$$

For our case (two modalities), again, the total objective is:

$$ELBO(x_a, x_b) + ELBO(x_a) + ELBO(x_b)$$

*2) Experiments:* We trained the model on the celebA dataset, with the first modality as the images and the second modality as feature vectors. We observed that there was major improvement in the quality of faces outputted, both from random noise and from specified feature vectors (present in Appendix).

Unfortunately, we still have not been able to implement the model on the moving MNIST dataset. However, looking at the quality of the samples from the celebA dataset, it is likely that this model will have a respectable qualitative accuracy, comparable to that of Denton et al., 2018.

## REFERENCES


[1] Kingma, D.P. and Welling, M., 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

[2] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).

[3] Wu, M. and Goodman, N., 2018. Multimodal generative models for scalable weakly-supervised learning. In Advances in Neural Information Processing Systems (pp. 5575-5585).

[4] Denton, E. and Fergus, R., 2018. Stochastic video generation with a learned prior. arXiv preprint arXiv:1802.07687.

[5] Larsen, A.B.L., Sønderby, S.K., Larochelle, H. and Winther, O., 2015. Autoencoding beyond pixels using a learned similarity metric. arXiv preprint arXiv:1512.09300.

[6] Vondrick, C., Pirsiavash, H. and Torralba, A., 2016. Generating videos with scene dynamics. In Advances In Neural Information Processing Systems (pp. 613-621).

[7] Babaeizadeh, M., Finn, C., Erhan, D., Campbell, R.H. and Levine, S., 2017. Stochastic variational video prediction. arXiv preprint arXiv:1710.11252.

[8] Zhou, Y., Wang, Z., Fang, C., Bui, T. and Berg, T.L., 2018. Visual to sound: Generating natural sound for videos in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3550-3558).

[9] Ephrat, A., Halperin, T. and Peleg, S., 2017. Improved speech reconstruction from silent video. In Proceedings of the IEEE International Conference on Computer Vision (pp. 455-462).

[10] Kingma, D.P. and Dhariwal, P., 2018. Glow: Generative flow with invertible 1x1 convolutions. In Advances in Neural Information Processing Systems (pp. 10215-10224).

[11] Dinh, L., Sohl-Dickstein, J. and Bengio, S., 2016. Density estimation using real nvp. arXiv preprint arXiv:1605.08803.

[12] Dinh, L., Krueger, D. and Bengio, Y., 2014. Nice: Non-linear independent components estimation. arXiv preprint arXiv:1410.8516.

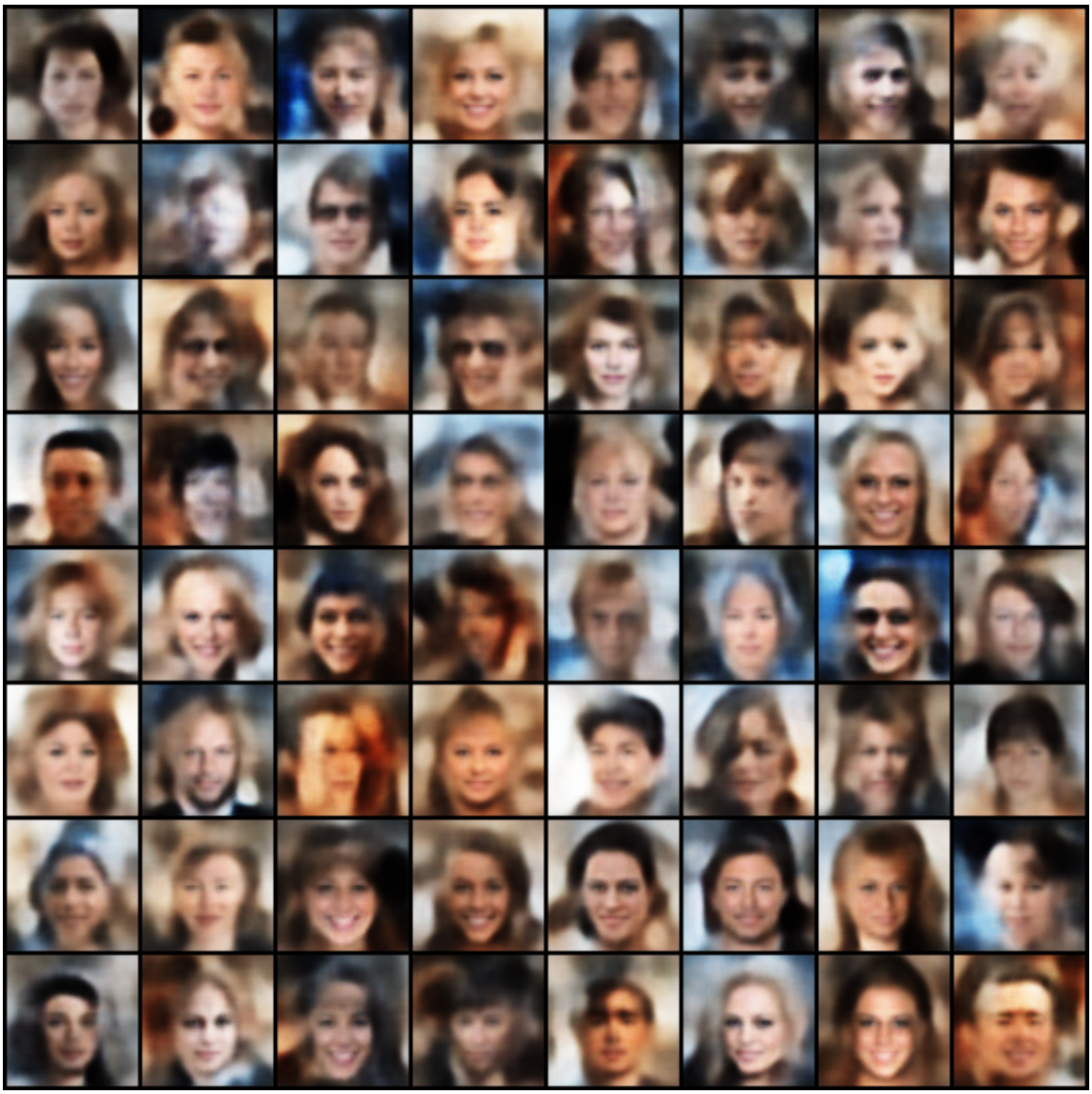

Fig. 9. Images constructed from random noise in the first proposed model.

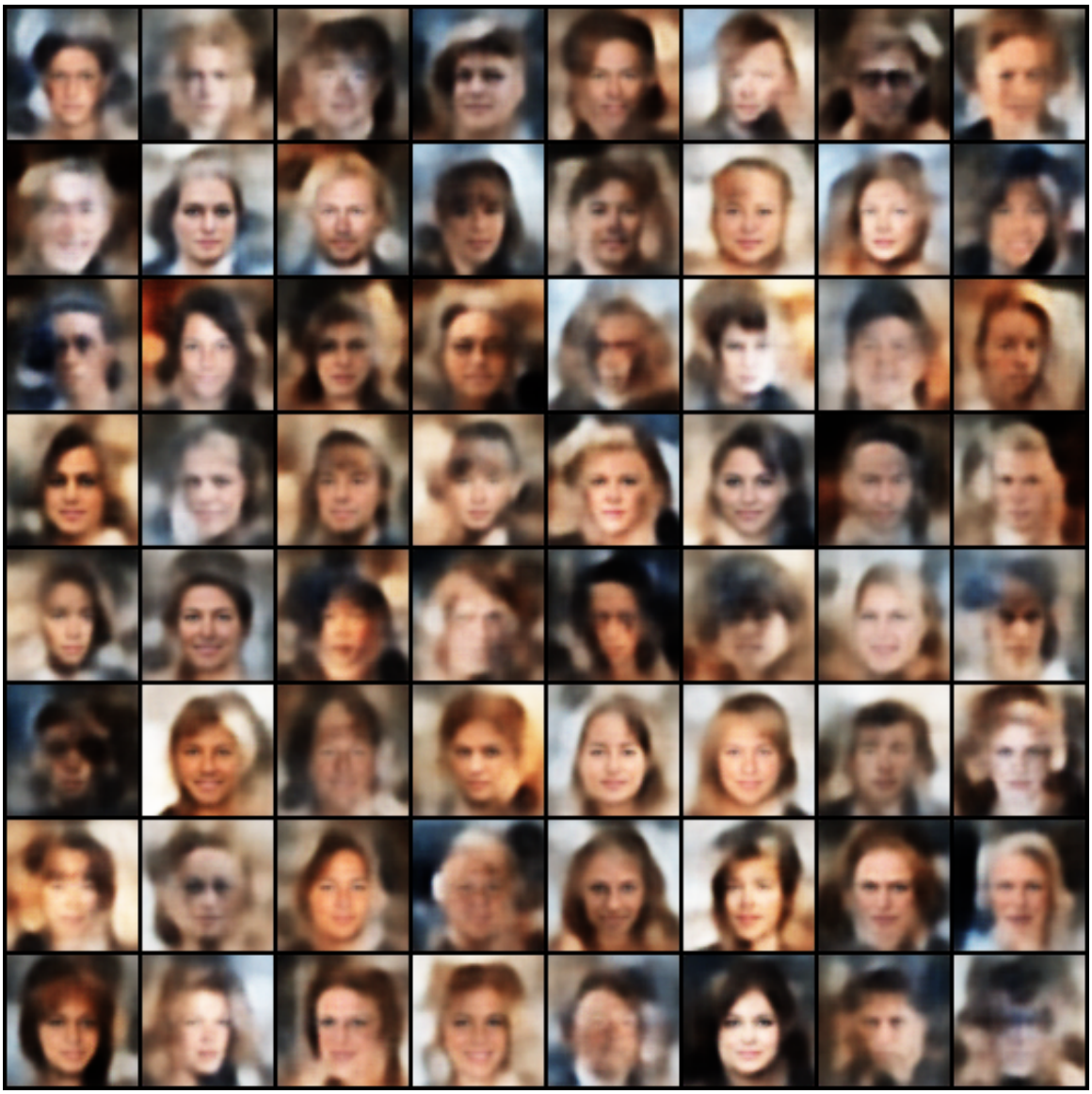

Fig. 10. Images constructed with (attribute = male) in the first proposed model.

Fig. 11. Images constructed with (attribute = male) in the second proposed model. Note that this model was not be trained till the losses were minimised, so there is still room for improvement.