# Adversarial Variational Bayes Generative Translation Network

**Michael Griffin**
mgriff94@utexas.edu

**Sam Kanawati**
omk09@utexas.edu

**Tao Zhang**
taozhang.zhg@utexas.edu

## Abstract

Unsupervised image-to-image translation aims at learning a joint distribution of images in different domains by using images from the marginal distributions in individual domains. In this paper, we adopt a prior work that proposes an unsupervised image-to-image translation framework called UNIT GAN that is based on learning a shared latent space of the two domains using a VAE and GAN style network. Our contribution relies on introducing Adversarial Variational Bayes which is technique for training Variational Autoencoders with an arbitrarily expressive posterior model. This approach encodes the images into a much expressive latent variable models that can be used to learn complex probability distributions from training data. We compare the proposed framework with the vanilla UNIT GAN baseline and show that our model converges in orders of magnitude less training epochs.

## 1 Introduction

Many researchers have leveraged adversarial learning for image-to-image translation, whose goal is to translate an input image from one domain to another domain given input-output image pairs as training data. One of the first such models, pix2pix [6], uses a conditional generative adversarial network (cGAN) [15] to learn a mapping from an input image to an output image. One application was image colorizing, were the input image is a black and white and the target image is the colored version. The generator in this case is trying to learn how to colorize a black and white image. The discriminator looks at the generator's colorization attempts and trying to learn to tell the difference between the colorizations the generator provides, and the true colorized target image provided in the dataset.

While pix2pix can produce pretty decent results, the challenge is in both the training data and the deterministic nature of the model. The training set of the two image domains that the model will learn to translate between needs to be aligned image pairs. For example, if we are learning to map night images to day images, our training set should be a set of paired images where each pair is the exact image taken in both during day and night. Furthermore, according to the authors, pix2pix trains a conditional GAN in a way that doesn't produce highly stochastic output (only test time Dropout as a form of stochasticity), thus the generator doesn't capture the full entropy of the conditional distribution modeled and isn't able to produce different image translations.

Some work has to tried to address the issue of having a paired dataset. CycleGAN [16] and later DiscoGAN [7] learn this mapping between different domains when given unaligned images. Further, these models also perform translation between domains in both directions as opposed to the unidirectional pix2pix model. However, they still don't produce more than one image per translation and aren't really generative models. This would be useful to see different possible translations of an image to another domain. One could consider several possible mappings of a black and white image to a colorized image as described above.

In this paper, we discuss a general framework for unsupervised image to image translation that is more generative than the existing works and does so in a more principled way than recent work (we consider the new UNIT architecture [10] discussed below). We present the architecture and discuss

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

how it learns to translate an image from one domain to another without any corresponding images in two domains in the training dataset. Furthermore, we explore a new training procedure based on using Adversarial Variational Bayes [14] based on adversarial training that allows us to make the inference model much more flexible, effectively allowing it to represent almost any family of conditional distributions over the latent variables.

## 2 Related Work

Our work is primarily concerned with adapting two recent works. We look at UNIT, a generative model for doing translation between different unpaired image domains. We also discuss Adversarial Variational Bayes and it's improvement upon standard Variational Autoencoders.

### 2.1 UNIT: Unsupervised Image-to-Image Translation Networks

The key challenge when approaching this problem from a probabilistic modeling perspective is to learn a joint distribution of images in different domains. The two sets consist of images from two marginal distributions in two different domains, and the task is to infer the joint distribution using these images. Although there exist an infinite set of joint distributions that can arrive the given marginal distributions in general the paper proposes to add additional assumptions on the structure of the joint distribution. The authors propose a shared-latent space assumption, which assumes a pair of corresponding images in different domains can be mapped to a same latent representation in a shared-latent space.

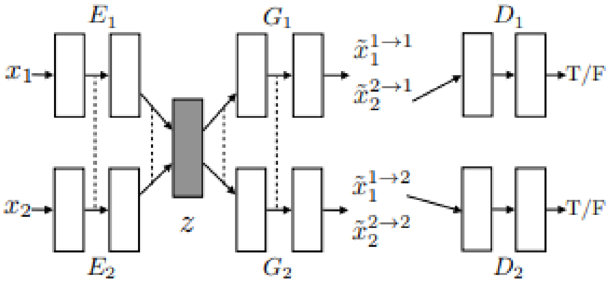

Figure 1: UNIT architecture with encoders $E_1$ and $E_2$, generators $G_1$ and $G_2$, discriminators $D1$ and $D2$, and image domains $X_1$ and $X_2$

The UNIT framework is based on combining generative adversarial networks (GANs) and variational autoencoders (VAEs) in a multi-modal way. Looking at Figure 1 we see that the model is composed of several interesting sub-parts. $E_1$ and $G_1$ form a VAE for domain $X_1$. $E_1$ and $G_2$ represent a standard image translation framework, similar to pix2pix, albeit in a VAE style generative fashion. $G_1$ and $D_1$ are a typical GAN where $G_1$ tries to produce more and more realistic results and $D_1$ tries to discriminate between those generated and actual images in domain $X_1$. $E_1$, $G_1$, and $D1$ combine to form a VAE-GAN [9] which is a method for training VAE's with a learned reconstruction loss function. Note all the previous models are similar for the other domain $X_2$. And finally $G_1$, $G_2$, $D1$, and $D2$ forms a recent architecture called CoGAN [11] which is a form of GAN which produces images in multiple domains from one shared latent variable $z$. One interesting implementation detail from their architecture was the decision to use shared layers in the encoder and decoders to help learn this shared representation.

UNIT accomplished the training of this complex architecture by a combination of loss functions for standard GAN's, VAE's, and also enforcing the cycle consistency loss found in the CycleGAN paper. Below, we see the loss functions that UNIT uses. Of particular note, they use a Laplace likelihood function for the generators $G_1$ and $G_2$, so when considering the log-likelihood reconstruction error, they use a standard $L_1$ loss function. Additionally, they use a simple posterior function in which the encoders $E_1$ and $E_2$ output the mean to a unit variance Gaussian distribution. Through the re-parameterization trick from VAE's, they then sample a latent variable $z$ to then run through the generators $G_1$ and $G_2$. Seen below are the loss functions used to train UNIT.

$$
\begin{aligned}
\min_{E_1,E_2,G_1,G_2}\ \max_{D_1,D_2}\ & \mathcal{L}_{\text{VAE}_1}(E_1,G_1) + \mathcal{L}_{\text{GAN}_1}(E_1,G_1,D_1) + \mathcal{L}_{\text{CC}_1}(E_1,G_1,E_2,G_2) \\
& + \mathcal{L}_{\text{VAE}_2}(E_2,G_2) + \mathcal{L}_{\text{GAN}_2}(E_2,G_2,D_2) + \mathcal{L}_{\text{CC}_2}(E_2,G_2,E_1,G_1). \qquad (1)
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{L}_{\text{VAE}_1}(E_1,G_1) =& \lambda_1 \text{KL}(q_1(z_1|x_1)||p_\eta(z)) - \lambda_2 \mathbb{E}_{z_1 \sim q_1(z_1|x_1)}[\log p_{G_1}(x_1|z_1)] \qquad (2) \\
\mathcal{L}_{\text{VAE}_2}(E_2,G_2) =& \lambda_1 \text{KL}(q_2(z_2|x_2)||p_\eta(z)) - \lambda_2 \mathbb{E}_{z_2 \sim q_2(z_2|x_2)}[\log p_{G_2}(x_2|z_2)]. \qquad (3)
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{L}_{\text{GAN}_1}(E_1,G_1,D_1) &= \lambda_0 \mathbb{E}_{x_1 \sim P_{\mathcal{X}_1}}[\log D_1(x_1)] + \lambda_0 \mathbb{E}_{z_2 \sim q_2(z_2|x_2)}[\log(1 - D_1(G_1(z_2)))] \qquad (4) \\
\mathcal{L}_{\text{GAN}_2}(E_2,G_2,D_2) &= \lambda_0 \mathbb{E}_{x_2 \sim P_{\mathcal{X}_2}}[\log D_2(x_2)] + \lambda_0 \mathbb{E}_{z_1 \sim q_1(z_1|x_1)}[\log(1 - D_2(G_2(z_1)))]. \qquad (5)
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{L}_{\text{CC}_1}(E_1,G_1,E_2,G_2) =& \lambda_3 \text{KL}(q_1(z_1|x_1)||p_\eta(z)) + \lambda_3 \text{KL}(q_2(z_2|x_1^{1\to2}))||p_\eta(z)) - \\
& \lambda_4 \mathbb{E}_{z_2 \sim q_2(z_2|x_1^{1\to2})}[\log p_{G_1}(x_1|z_2)] \qquad (6) \\
\mathcal{L}_{\text{CC}_2}(E_2,G_2,E_1,G_1) =& \lambda_3 \text{KL}(q_2(z_2|x_2)||p_\eta(z)) + \lambda_3 \text{KL}(q_1(z_1|x_2^{2\to1}))||p_\eta(z)) - \\
& \lambda_4 \mathbb{E}_{z_1 \sim q_1(z_1|x_2^{2\to1})}[\log p_{G_2}(x_2|z_1)]. \qquad (7)
\end{aligned}
$$

For sake of brevity, we will refer you to the original papers for GAN's [3], VAE's [8], and CycleGAN [16] as to the derivation of these loss functions. Briefly, the GAN loss function ensures that UNIT generates realistic images from it's generators, the VAE loss ensures that the posterior matches the prior distribution (in this case a simple 0 mean, unit variance Normal distribution) while minimizing reconstruction error, and the cycle loss function ensures that when an image is translated to another domain and back, that the resulting image should be close to the original image.

## 2.2 AVB: Adversarial Variational Bayes

The failure of VAEs to generate sharp images is often attributed to the fact that the inference models used during training are usually not expressive enough to capture the true posterior distribution. Adversarial Variational Bayes, is a technique for training Variational Autoencoders with arbitrarily flexible inference models parameterized by neural networks. The Kullback-Leibler regularization term that appears in the training objective for VAE's is replaced with an adversarial loss that encourages the aggregated posterior to be close to the prior over the latent variables. Unlike Adversarial Autoencoders [13], AVB instead uses an arbitrarily complex posterior function where the noise from the re-parameterization trick is injected into the input rather than when sampling from the posterior for $z$ as can be seen in Figure 2 below.

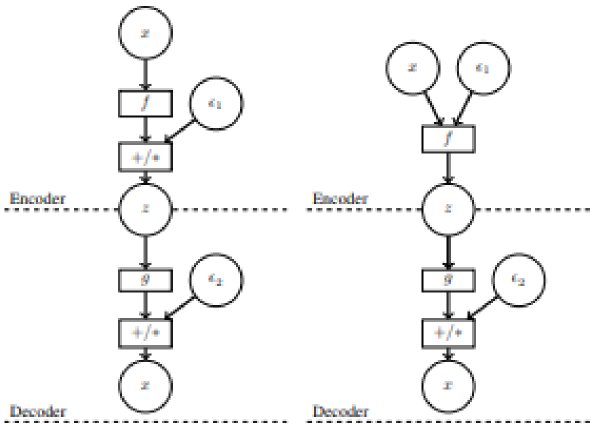

Figure 2: Left: Standard VAE with noise injected in the middle for simple Gaussian posterior. Right: AVB with noise injected in the input for arbitrarily complex posterior functions

This allows the model to map the image to a latent variable $z$ using a posterior which approximates the true data driven posterior function, rather than a simple Gaussian parameterized posterior function. The idea of the AVB approach is to introduce a discriminative network $T(x, z)$ that tries to distinguish pairs $(x, z)$ that were sampled independently using the distribution $p_D(x), p(z)$ from those that were sampled using the current inference model, using $p_D(x), q(z|x)$. The discriminator is considered to be a nonparametric limit and a universal function approximator to approximate the Kullback-Leibler divergence term. We refer the reader to the AVB paper [14] for the proof of this.

The loss function for this model changes from the standard VAE loss function to the ones below. Note, we use the negative output of the discriminator $T(x, z)$ in place of the KL term in formulating the standard ELBO loss function. $T(x, z)$ is then trained using a standard GAN discriminator loss as seen below also.

$$\max_{\theta,\phi} \mathbb{E}_{p_{\mathcal{D}}(x)} \mathbb{E}_{\epsilon} \big( -T^{*}(x, z_\phi(x,\epsilon)) + \log p_\theta(x \mid z_\phi(x,\epsilon)) \big). \tag{8}$$

$$\max_{T} \mathbb{E}_{p_{\mathcal{D}}(x)} \mathbb{E}_{q_\phi(z|x)} \log \sigma(T(x,z)) + \mathbb{E}_{p_{\mathcal{D}}(x)} \mathbb{E}_{p(z)} \log\left(1 - \sigma(T(x,z))\right). \tag{9}$$

## 3 AVB-GTN: Adversarial Variational Bayes Generative Translation Network

Our contribution comes from incorporating both of the previous architectures, UNIT and AVB, into one model. This allows us to not only create a generative multi-modal translation network similar to UNIT, but to do so in way that maintains a better latent representation of the input images. Since UNIT uses a simple unit variance posterior function, it is limited in it's capacity to represent an image as a latent variable $z$ as discussed above with standard VAE's. This not only limits the ability of the network to map images to a shared latent code, but also in reconstructing an image from said latent code. A poor latent representation of an image won't produce nearly as good output images (as we will show later through our experiments).

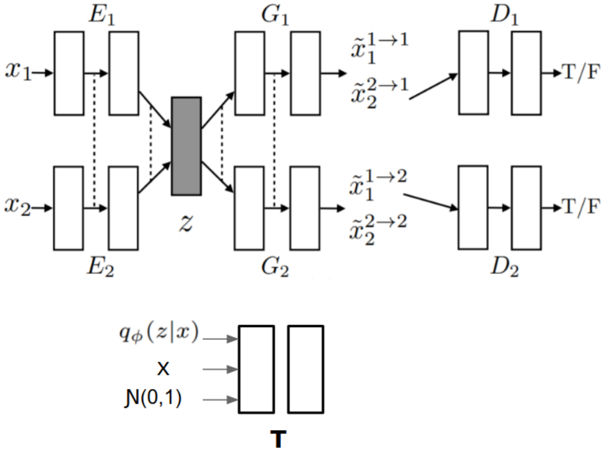

Figure 3: Our model which uses a standard UNIT architecture along with a AVB style discriminator.

As you can see above in Figure 3, our model is a rather simple addition to the standard UNIT architecture. The notable addition is that of a discriminator (akin to that of AVB) which learns to discriminate between latent codes produce from the encoder (the posterior function) and latent codes sampled from the prior distribution. This replaces the Kullback-Leibler divergence terms in the UNIT loss functions with the output of the discriminator $T$ as in AVB. We then add an additional loss function which trains the discriminator, exactly as in AVB. In this way, the generator is trained in almost the exact same way and the discriminator is left untouched. The loss functions for the new architecture are shown here:

$$
\begin{aligned}
\mathcal{L}_{\text{VAE}_1}(E_1, G_1) =& \mathbb{E}_{x_1 \sim P_{\mathcal{X}_1}} \mathbb{E}_{z_1 \sim q_1(z_1|x_1)}[\lambda_1 \log \sigma(T(x_1, z_1)) - \lambda_2 \log p_{G_1}(x_1|z_1)] \qquad (10)\\
\mathcal{L}_{\text{VAE}_2}(E_2, G_2) =& \mathbb{E}_{x_2 \sim P_{\mathcal{X}_2}} \mathbb{E}_{z_2 \sim q_2(z_2|x_2)}[\lambda_1 \log \sigma(T(x_2, z_2)) - \lambda_2 \log p_{G_2}(x_2|z_2)]. \qquad (11)
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{L}_{\text{CC}_1}(E_1, G_1, E_2, G_2) =& \lambda_3 \mathbb{E}_{x_1 \sim P_{\mathcal{X}_1}} \mathbb{E}_{z_1 \sim q_1(z_1|x_1)}[\log \sigma(T(x_1, z_1))] + \lambda_3 \mathbb{E}_{z_2 \sim q_2(z_2|x_1^{1\to 2})}[\log \sigma(T(x_1^{1\to 2}, z_2))]\\
&- \lambda_4 \mathbb{E}_{z_2 \sim q_2(z_2|x_1^{1\to 2})}[\log p_{G_1}(x_1|z_2)] \qquad (12)\\
\mathcal{L}_{\text{CC}_2}(E_1, G_1, E_2, G_2) =& \lambda_3 \mathbb{E}_{x_2 \sim P_{\mathcal{X}_2}} \mathbb{E}_{z_2 \sim q_2(z_2|x_2)}[\log \sigma(T(x_2, z_2))] + \lambda_3 \mathbb{E}_{z_1 \sim q_1(z_1|x_2^{2\to 1})}[\log \sigma(T(x_2^{2\to 1}, z_1))]\\
&- \lambda_4 \mathbb{E}_{z_1 \sim q_1(z_1|x_2^{2\to 1})}[\log p_{G_2}(x_2|z_1)] \qquad (13)
\end{aligned}
$$

Note that these loss functions are nearly identical to those of the UNIT loss functions, albeit with the KL terms replaced with our new discriminator. The existing GAN loss remains the same and we use the same loss function to train the discriminator $T$ as AVB.

## 4 Experiments

We experiment our framework on two tasks: image-to-image transformation and audio-to-audio transformation. Image-to-image transformation is trained and tested on CelebA dataset [12], and we focus on transforming the hair color from brown to blond and the other way around. Our result is compared with those generated by UNIT. For the audio-to-audio transformation, the two domains in our GAN architecture are stereo music played by violin and piano from the IRMAS dataset [2].

### 4.1 Implementation

For our models we chose to use architectures similar to what they used in the original UNIT paper. We used a ResNet style image encoder and decoder with shared middle layers before and after the latent code, discriminators with a shared output, and a KL discriminator with a simple CNN style architecture. When doing experiments using the original UNIT code as well, we used the exact same architecture except for our additional KL discriminator so as to establish a suitable baseline. We trained for a maximum of 50,000 epochs with a batch-size of 1 on all experiments due to GPU hardware constraints.

### 4.2 Image-to-Image Transformation

We experiment our architecture on CelebA dataset. Figure 4 shows the transformation of hair color for one image from the dataset. As shown in Figure 4 (a), our GAN is able to successfully convert from blond hair color into brown. Compared with the result from UNIT, our result has much less deformation of the celebrity's face while maintaining good color transformation. Besides, our brown hair color looks more natural while UNIT generates hair that looks transparent in some part, which is not natural. For the conversion of brown into blond, which is shown in Figure 4 (b), we also have better result as compared to UNIT in terms of less face deformation and being more natural. In both cases, our model has demonstrated better results. This is due to the changes we made , which enables more expressive latent space while UNIT adopts the same encoder structure as a standard VAE. It can also seen in Figure 5 that our model also generalizes well to faces not in the dataset. Finally, in Figure 6 we can that we still get slightly better results than the original UNIT paper but with many orders of magnitude less epochs.

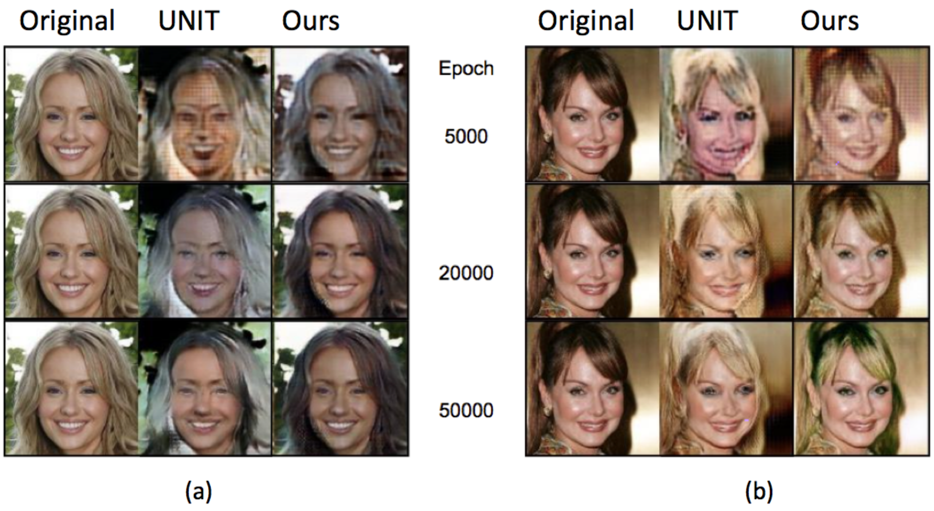

Figure 4: Hair color transformation vs. training progress: (a) blond to brown, and (b) brown to blond. The Vertical axis denotes the number of Epoch of training.

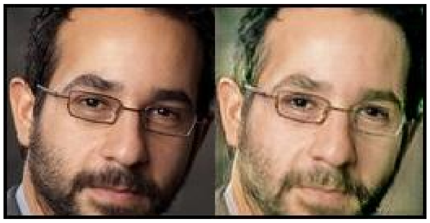

Figure 5: An example of going from brown to blond hair on an image outside of the dataset.

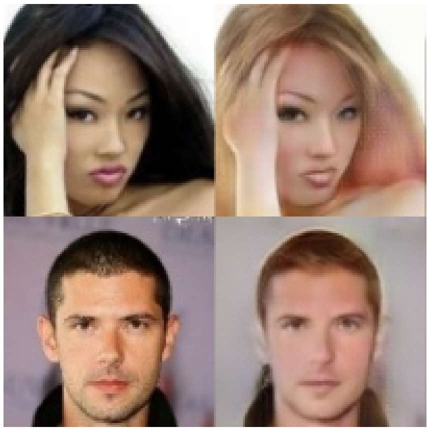

Figure 6: Image translation results from the original UNIT paper after training for 2,000,000 epochs. You can see that the faces are blurred quite a bit as well.

### 4.3 Audio-to-Audio Transformation

In this part, we apply our GAN into transforming audio signals between those played by violin and piano. In order to perform such transformation, we first convert audios into spectrogram images via Short Time FFT (STFT), based on which we train our model. Then we convert test audios into spectrogram images via STFT, generate new spectrogram images with our GAN, and finally convert them back into a new audio using Inverse STFT. Though it is known that we can reconstruct from the magnitude-only spectrogram [4] , we find during our experiment that quality of such reconstruction deteriorates when we down-sample the spectrogram images to accelerate the STFTs. As a result, in our experiment we maintain both magnitude and phase information in the frequency domain of the audio signals, which is different from previous work [1] [5] .

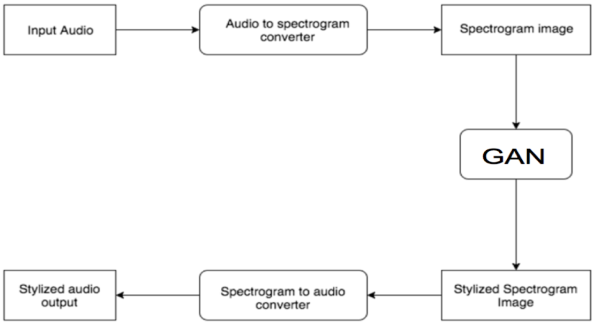
Input Audio
Audio to spectrogram converter
Spectrogram image
GAN
Stylized Spectrogram Image
Spectrogram to audio converter
Stylized audio output

Figure 7: Architecture of the audio-to-auto transformation system.

We experiment on violin-to-violin, violin-to-piano, violin-to-violin-to-piano, piano-to-piano, piano-to-violin, and piano-to-violin-to-piano transformations, and the spectrograms can be seen in 8. One can observe that piano-violin and violin-to-piano has similar spectrogram magnitudes, which is reasonable because they can both be seen as mixture of piano and violin audios. Similarly, the audio transformation between the same domains (i.e. violin-to-violin and piano-to-piano) also looks similar. Also, from the violin and piano spectrograms, we can see that violin has higher pitch in this audio, which is consistent with what we hear from the audio files.

After we convert theses spectrograms into audios, we can recognize the mixture of different instruments but the audio quality is very poor. This might be caused by the improper generated phase. As can be seen from 9, the phase changes very fast and almost covers the whole image, which make it different for our GAN to capture useful information from it. Out future study should focus on how to extract useful information from phase image.

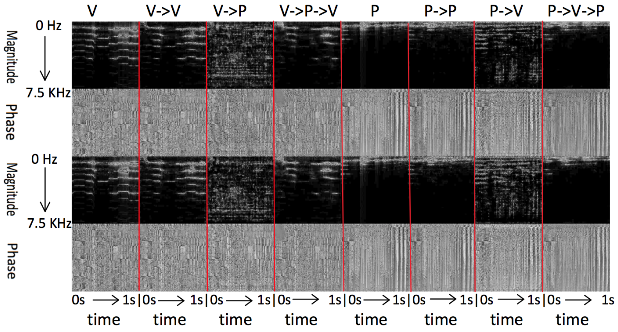

Figure 8: Spectrograms of the audio signals. The upper two rows are magnitudes and phases of the left channel of the stereo audios, while the lower two are right channel. Each column represents spectrogram from one single audio. Note that the 'P' and 'V' respectively denotes spectrogram of the the original audio by piano and violin, while 'P->P' denotes the transformed spectrogram of piano to piano, the other notations are similar. In order for better visibility, the magnitudes are multiplied by 10.

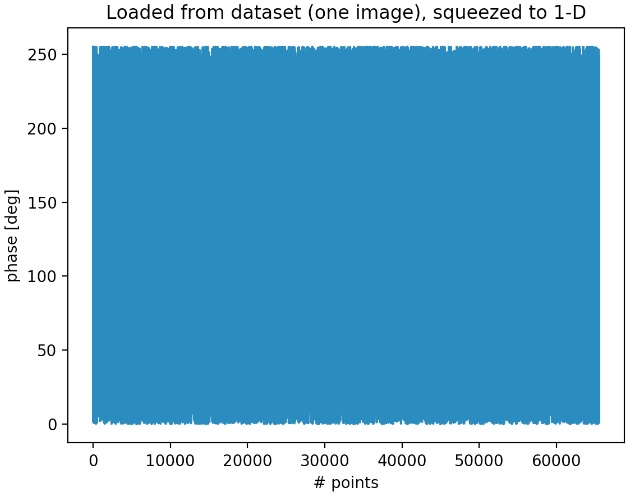

Figure 9: Phase of one typical audio from the dataset. Note that the 2-D matrix has been squeezed into 1-D for visualization, and it has been normalized to 0 255.

## 5 Conclusion

We presented a general framework for Adversarial Variational Bayes Generative Translation Network. We showed how it learned to translate an image from one domain and does so much faster than previous works. Our framework adopts a new training procedure for Variational Autoencoders based on Adversarial Variational Bayes training. This allows us to make the posterior model much more flexible, effectively allowing it to represent almost any family of conditional distributions over the latent variables.

## References

[1] Audiostyletransfer. `http://gauthamzz.com/2017/09/23/AudioStyleTransfer/`. Accessed: 2018-05-04.

[2] J. J. Bosch, J. Janer, F. Fuhrmann, and P. Herrera. A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals. *ISMIR*, 2012.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[4] Daniel Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.

[5] Eric Grinstein, Ngoc Q. K. Duong, Alexey Ozerov, and Patrick Pérez. Audio style transfer. *CoRR*, abs/1710.11385, 2017.

[6] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016.

[7] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. *CoRR*, abs/1703.05192, 2017.

[8] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[9] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *CoRR*, abs/1512.09300, 2015.

[10] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. *CoRR*, abs/1703.00848, 2017.

[11] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 469–477. Curran Associates, Inc., 2016.

[12] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[13] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. Adversarial autoencoders. In *International Conference on Learning Representations*, 2016.

[14] Lars M. Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. *CoRR*, abs/1701.04722, 2017.

[15] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

[16] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017.