LATENT DOMAIN TRANSFER: CROSSING MODALITIES WITH BRIDGING AUTOENCODERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Domain transfer is an exciting and challenging branch of machine learning because models must learn to smoothly transfer between domains, preserving local variations and capturing many aspects of variation without labels. However, most successful applications to date require the two domains to be closely related (e.g., image-to-image, video-video), utilizing similar or shared networks to transform domain-specific properties like texture, coloring, and line shapes. Here, we demonstrate that it is possible to transfer across modalities (e.g., image-to-audio) by first abstracting the data with latent generative models and then learning transformations between latent spaces. We find that a simple variational autoencoder is able to learn a shared latent space to bridge between two generative models in an unsupervised fashion, and even between different types of models (e.g., variational autoencoder and a generative adversarial network). We can further impose desired semantic alignment of attributes with a linear classifier in the shared latent space. The proposed variation autoencoder enables preserving both locality and semantic alignment through the transfer process, as shown in the qualitative and quantitative evaluations. Finally, the hierarchical structure decouples the cost of training the base generative models and semantic alignments, enabling computationally efficient and data efficient retraining of personalized mapping functions.

1 INTRODUCTION

Domain transfer has long captured the imagination of inventors and artists alike. The early precursor of the phonograph, the phonautograph, was actually inspired by the idea of "words which write themselves", where the shape of audio waveforms would transform into the shape of writing, capturing the content and character of the speaker's voice in shape and stroke of the written characters (Feaster, 2009). While perhaps fanciful at the time, modern deep learning techniques have shown similar complex transformations are indeed possible and exciting.

Deep learning largely enables transfer through providing a mapping between two domains such that the variations in one domain are reflected in the other. Interesting existing works include transferring between two domains of different styles of image (Isola et al., 2016; Zhu et al., 2017; Li et al., 2018; Li, 2018), video (Wang et al., 2018) and music (Mor et al., 2018), and are usually referred as "style transfer" since they are operating on the style level. These works, besides being visually appealing, allows intuitive control of complicated generative models that users can expect the result in the target domain is intuitively related to the source domain. Furthermore, learning transformation is easier than learning the full generative model.

Despite being interesting, this line of works in domain transfer has several limitations. The first limitation is that it requires that two domains where the model transfers should be closely related (e.g. image-to-image or video-to-video). The model also transfer local properties like texture and coloring instead of high-level semantics, for example, directly applying CycleGAN or its variants to images of not-so-close domains leads to distorted or wired results (Li et al., 2018). Even the task of transferring between texture styles are not done principally, as shown in Chu et al. (2017) that CycleGAN hides texture on source domain in the high-frequency part of target domain and generate the visible texture (low-frequency part) regardless of the source.

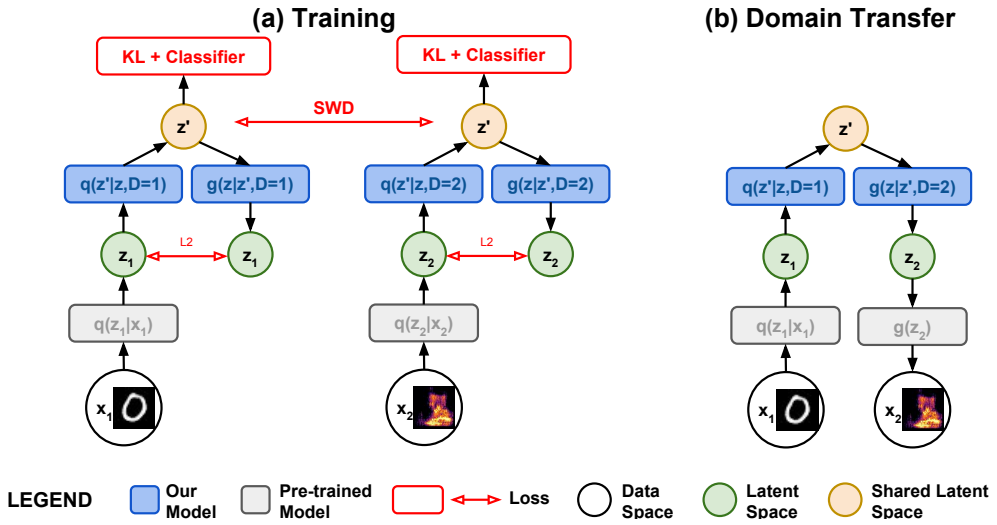

Figure 1: Architecture and training. Our method aims at transfer from one domain to another domain such that the correct semantics (e.g., label) is maintained across transferring and local changes in source domain should be reflected in the target domain. It operates on the latent spaces of pre-trained generative models on source and target domain respectively. **(a)** The training is done with losses that fall into three categories: (1) The VAE losses to encourage modeling of z_1 and z_2, which are denoted as L2 and KL in the figure. (2) The Sliced Wasserstein Distance loss to encourage cross-domain overlapping in the shared latent space, which is denoted as SWD. (3) The classification loss to encourage a-class overlapping in the shared latent space, which is denoted as Classifier. The training is semi-supervised, since (1) and (2) requires no supervision (classes) while only (3) needs such information. **(b)** To transfer data from one domain x_1 (an image of digit "0") to another domain x_2 (an audio of human saying "zero", shown in form of spectrum in the example), we first get encoding of x_1 by $z_1 \sim q(z_1|x_1)$, followed by getting the shared encoding of z_1 using our conditional encoder by $z' \sim q(z'|z_1, D = 1)$ where D donates the operating domain. Then we get encoding in the target domain $z_2 = g(z|z', D = 2)$ using our conditional decoder, which is fed into pre-trained generator $x_2 = g(x_2|z_2)$.

The second limitation is data efficiency: Some techniques in the line of style transfer need dense supervision, which means lots of paired data. For example, vid2vid (Wang et al., 2018) requires frame-by-frame pairs as the training data. Usually one side of the domains essentially a dense feature extraction of the other, for example in vid2vid's case one side is the segmentation of the other as an image. This makes the transfer model learn an inverse feature extraction, and is usually infeasible for different domains, such as image-to-audio scenario.

To tackle prior limitations, we propose a new approach that enables transfer from a source domain to a target domain, that has two properties: (1) *locality*, which means the local variations in source domain should be reflected in the target domain, and (2) *semantic alignment*, which means the model correct semantics are shared between source and target domain through domain, where the semantic can be one of many possible valid ones. Our proposed method satisfies both properties and enables the cross-domain transfer where domains are possible drastically different.

Our main contributions include:

- We propose to tackle the domain transfer through a conditional VAE on top of latent spaces of pre-trained generative models of source and target domain, trained in a semi-supervised way. The new VAE has a higher shared latent space, and are trained jointly with cross-domain Wasserstein distance and semantic classification loss in that space.
- Our proposed method enables multiple settings of transfer, including images of digits to images of digits, images of digits to images to fashion items, and even images of digits to audio of digits.

- We show both qualitative and quantitative results for transferring in multiple settings of transfers, which demonstrates that our proposed methods for transfer can produce visually appealing results satisfying both *locality* and *semantic alignment*, plus have high accuracy in maintaining semantics through subjective measurement.
- Our proposed method demonstrates efficiency in training, measured by both required number of supervision as well as clock time.

2 METHOD

2.1 PROBLEM DEFINITION

We seek to train a model capable of transferring instances from a source domain (x_1) to a target domain (x_2), such that local variations in source domain are transferred to local variations in the target domain. We refer to this property as *locality*. Thus, local interpolation in the source domain would ideally be similar to local interpolation in target domain when transferred.

There are many possible ways that two domains could align such that they maintain locality, with many different alignments of semantic attributes. For instance, for a limited dataset, there is no a priori reason that images of the digit seven and spoken utterances of the digit seven would align to each other. Or more abstractly, there may be no agreed common semantics for images of landscapes and passages of music, and it is at the liberty of the user to define such connections based on their own intent. Our goal in modeling is to respect the user's intent and make sure that the correct semantics (e.g., labels) are shared between the two domains during transfer. We refer to this property as *semantic alignment*. A user can thus sort a set of data points from in each domain into common bins, which we can use to constrain the cross-domain alignment. We can quantitatively measure the degree of semantic alignment by using a classifier to label transformed data and measuring the percentage of data points that fall into the same bin for the source and target domain.

Our goal can thus be stated as learning transformations that preserve locality and semantic alignment, while requiring as few labels from a user as possible.

2.2 ARCHITECTURE

Figure 1 diagrams our architectural approach solving this problem. We first independently pre-train a separate generative model, either a VAE or GAN, for both the source and target domain respectively.

For VAEs, data is encoded from the data domain to the latent space through a learned encoding function ($z \sim q(z|x)$)), and decoded back to the data space with a decoder function (($\hat{x} \sim g(x|z)$)). For GANs, we choose latent samples from a spherical Gaussian prior $z \sim p(z)$ and then use rejection sampling to only select latent samples whose associated data $x = g(z)$ is classified with high confidence by an auxiliary classifier.

Then we add an additional level of hierarchy by adding a conditional VAE with shared weights that models the *latent spaces* z_1 *and* z_2 themselves. The VAE learns a higher latent space z', That we call a "shared latent space" since it corresponds to both domains. Sharing the weights encourages the model to seek common structures between the latent domains, but we also find it helpful to condition both the encoder $q(z'|z, D)$ and decoder $g(z|z', D)$, with an additional one-hot domain label, D, to allow the model some flexibility to adapt to variations particular to each domain.

The training of our proposed conditional VAE is done by optimizing against a full loss consisting of three categories of loss terms we describe as follows. (We denote approximated posterior $Z'_d \triangleq q(z'|z_d, D = d), z_d \sim q(z_d|x_d), x_d \sim p(x_d)$ for $d \in \{1, 2\}$, the process of sampling z'_d from domain d) **(1) VAE Loss** For each domain $d \in \{1, 2\}$, the VAE is fit to data to maximize the Evidence Lower Bound (ELBO) $\mathcal{L}_d^{\text{ELBO}} = \mathbb{E}_{z' \sim Z'_d} [\log \pi(z_d; g(z', D = d)] - \beta_{\text{KL}} D_{\text{KL}} (q(z'|z_d, D = d) \| p(z'))$ where both q and g are fit to maximize $\mathcal{L}_d^{\text{ELBO}}$ and the likelihood $\pi(z; g)$ be the product of $\mathcal{N}(z; g, \sigma^2 I)$ with σ that effectively sets $\log \pi(z; g) = ||z - g||_2$ or L2 Loss. **(2) SWD Loss** We introduce SWD, or Sliced Wasserstein Distance (Bonneel et al., 2015) that measures the distance of two mini-batches of samples S'_1 and S'_2, where S'_1 is sampled from source domain $z'_1 \sim Z'_1$ and S'_1 from the target domain $z'_2 \sim Z'_d$, which is $\mathcal{L}^{\text{SWD}} = 1/|\Omega| \sum_{\omega \in \Omega} W_2^2 (\text{proj}(S'_1, \omega), \text{proj}(S'_2, \omega))$ where Ω is a set of random unit vectors, $\text{proj}(A, a)$ is the projection of A on vector a, and $W_2^2(A, B)$

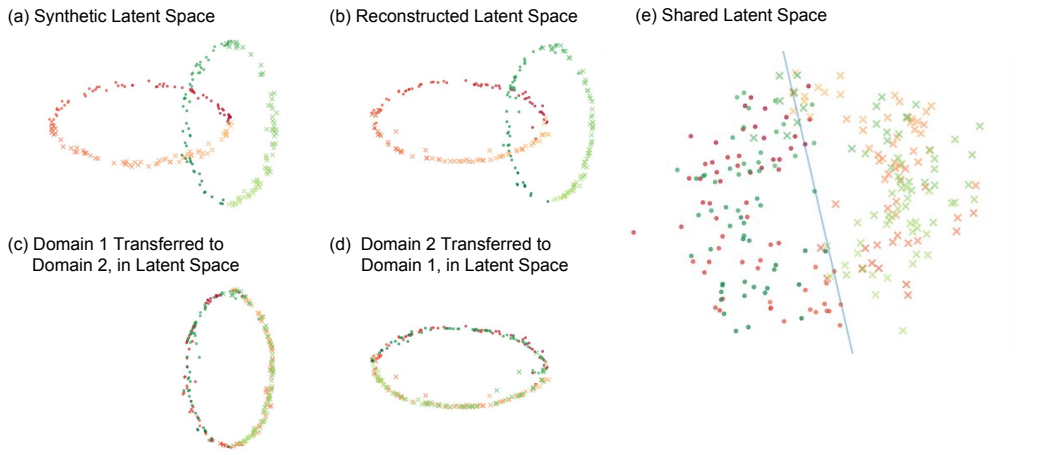

Figure 2: Synthetic data to demonstrate the transfer between 2-D latent spaces using our proposed conditional VAE. Shared latent space is also 2-D. Better viewed with color. **(a)** Synthetic data in latent space. The left eclipse denotes the domain 1's latent space, while the right eclipse domain 2's. The color spectrum denotes the continuity of local changes. There are two classes, A (denoted as dots) and B (denoted as crosses), for both domains. Also note that for domain 0, label A and B are arranged up-and-down, while for domain 1 left-and-right, which is intentionally designed to force the model to learn the rotation instead of "cheating" by squeezing the shape of ellipses. **(b)** Reconstructed latent space points using VAE. **(c)** Domain 1 transferred to domain 2. Note that the transferring correctly handles classes as well as continuity of local changes **(d)** Domain 2 transferred to domain 1. Observation is similar to (c). **(e)** The shared latent space where the blue line is the decision boundary of the classifier. Here, the points from both domains are overlapping, spreading evenly, and maintains the continuity of color spectrum. The class-level clustering can also be observed.

is the quadratic Wasserstein distance. **(3) Classification Loss** For both domain $d \in \{1, 2\}$ we introduce $\mathcal{L}_d^{\text{Cls}} = \mathbb{E}_{z' \in Z'_d} H(f(z'), l_{x'})$ where H is the cross entropy loss, $f(z')$ is a one-layer linear classifier, and $l_{x'}$ is the one-hot representation of label of x', the data associated with z'.

Putting all pieces together, the final loss is $\mathcal{L}_1^{\text{ELBO}} + \mathcal{L}_2^{\text{ELBO}} + \beta_{\text{SWD}}\mathcal{L}^{\text{SWD}} + \beta_{\text{Cls}}\mathcal{L}_1^{\text{Cls}} + \beta_{\text{Cls}}\mathcal{L}_2^{\text{Cls}}$. Using synthetic data, we show how our proposed method handles the desired transfer in Figure 2. We also detail the designing intuition behind loss terms in Appendix A.

2.3 Domain Reconstruction and Domain Transfer

Both domain reconstruction and domain transfer can be covered in the general scenario of going from domain d_1 to domain d_2, where for reconstruction $d_1 = d_2$ (e.g., $d_1 = d_2 = 2$) and for transfer $d_1 \neq d_2$ (e.g., $d_1 = 1, d_2 = 2$) . This general scenario is enabled by our proposed methods using the following process: Get z_{d_1}, the encoding of x_{d_1} in the latent space of pre-trained model on x_{d_1}, by sampling $z_{d_1} \sim q(z_{d_1}|x_{d_1})$; get z' in the shared latent space of the proposed conditional encoder, by sampling $z' \sim q(z'|z_{d_1}, D = d_1)$; get z_{d_2}, the encoder in the latent space of the pre-trained model on x_{d_2}, by computing $z_{d_2} = g(z|z', D = d_2)$ using the proposed conditional decoder; get data in the target domain by feeding z_{d_2} into pre-trained generator $x_{d_2} = g(x_{d_2}|z_{d_2})$.

3 Experiments

3.1 Datasets

While the end goal of our method is to enable creative mapping between datasets with arbitrary alignments, for quantitative studies we restrict ourselves to three domains where there exist a somewhat natural alignment that we can measure semantic alignment quantitatively:

1. **MNIST** (LeCun, 1998), which contains images of hand-written digits of 10 classes from "0" to "9".
2. **Fashion MNIST** (Xiao et al., 2017), which contains fashion related objects such as shoes, t-shirts, categorized into 10 classes. The structure of data and the size of images are identical to MNIST.
3. **SC09**, a subset of Speech Commands Dataset [1], which contains the record of audio of humans saying digits from "0" to "'9".

For MNIST and Fashion MNIST, we prepare VAE with MLP encoder and decoder following setting up in Engel et al. (2018). for SC09 we use the public available WaveGAN (Donahue et al., 2018)[2].

We set three scenarios of transferring:

1. MNIST $\leftrightarrow$ MNIST. In this scenario, the model is required to transfers between images a digit class to the same digit class, and a digit class should be transferred to the same digit class in the target domain.
2. MNIST $\leftrightarrow$ Fashion MNIST. In this scenario, the model is required to transfers between images of a class of digit and images of a class of fashion object, and we specify a global one-to-one mapping between 10 digit classes and 10 fashion object classes (See Table 5 in Appendix for details on this mapping);
3. MNIST $\leftrightarrow$ SC09. In this scenario, the model is required to transfer between images of a class of digit and audio of the same class of digits, and the image of one digit class transferred with same digit class' audio version, which means an image of hand-written "0" should go to an (arbitrary) audio version of "0" spoken by human.

We would like to emphasize that we only use *class level* supervision. Also specifically for MNIST $\leftrightarrow$ MNIST, we train two separate VAEs for MNIST with *different* random initialization with the intention to have two VAEs with different latent spaces pre-trained on the same data, which serves the role of demonstrating an upper-bound of our method.

The architecture of neural networks that implements the encoder q and decoder g in the conditional VAE is as follows: q and g are implemented using deep neural network. More specifically, we use stacks of fully-connected linear layers activated by ReLU, together with a "Gated Mixing Layer". Network architecture of conditional VAE is detailed in Appendix B

3.2 Domain Reconstruction and Domain Transfer

As mentioned in Section 2.3, both domain reconstruction and domain transfer are enabled as the process to transfer data from domain d_1 to domain d_2. We show two kinds of results, the qualitative one and the quantitative one. The qualitative results demonstrate visually the efforts of transformation, while the quantitative ones provide objective measurements. More specifically, the quantitative results are calculated with help of pre-trained classifier on domain d_1 and domain d_2, that is, when evaluating transferring from data x_{d_1} to x_{d_2}, the reported accuracy is the portion of instances that x_{d_1} and x_{d_2} have the same predicted classes.

For domain reconstruction, we select three settings to show the reconstruction as follows: (1) MNIST reconstruction in MNIST $\leftrightarrow$ MNIST setting, (2) Fashion MNIST reconstruction in MNIST $\leftrightarrow$ Fashion MNIST setting, and (3) SC09 reconstruction in MNIST $\leftrightarrow$ SC09 setting. The qualitative results are shown in Figure 3 and the quantitative results in form of reconstruction accuracy are shown in Table 1.

For domain transfer, we select three settings to show the transfer as follows: (1) MNIST $\leftrightarrow$ MNIST setting, (2) MNIST $\leftrightarrow$ Fashion MNIST setting, and (3) MNIST $\leftrightarrow$ SC09 setting. The qualitative results are shown in Figure 4 and the quantitative results in form of reconstruction accuracy are shown in Table 2.

[1] Available at `https://ai.googleblog.com/2017/08/launching-speech-commands-dataset.html`. License: `https://creativecommons.org/licenses/by/4.0/`

[2] Avaialble at `https://github.com/chrisdonahue/wavegan`

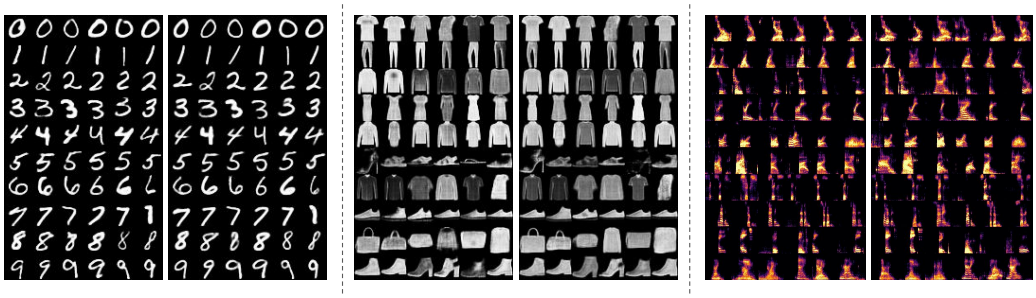

Figure 3: Qualitative Results for Domain Reconstruction. Images are divided into three groups, representing MNIST, Fashion MNIST and SC09 reconstruction respectively from left to right. Specifically, for SC09 we show the spectrum of audio. Within each group, on the left is the original and on the right reconstruction. This example shows that the reconstruction is of high-quality.

Data Domain	MNIST	Fashion MNIST	SC09
Accuracy	0.989	0.903	0.739

Table 1: Domain Reconstruction Accuracy, for MNIST, Fashion MNIST and SC09 reconstruction respectively.

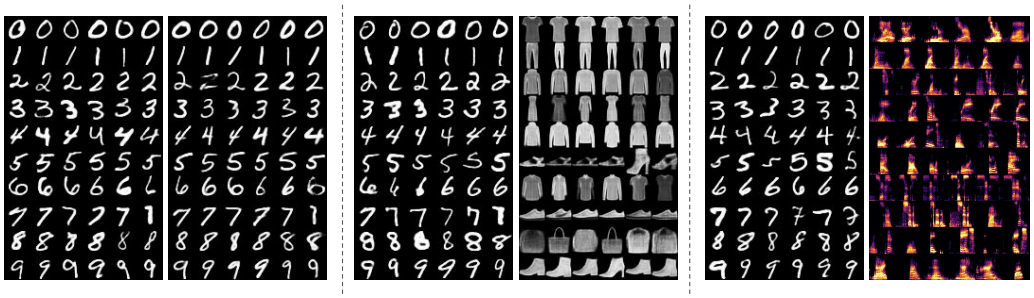

Figure 4: Qualitative Results for Domain Transfer. Images are divided into three groups, representing MNIST $\leftrightarrow$ MNIST , MNIST $\leftrightarrow$ Fashion MNIST and MNIST $\leftrightarrow$ SC09 transfer respectively from left to right. Specifically, for SC09 we show the spectrum of audio. Within each group, on the left is the data in the source domain and on the right is the data in the target domain. This example shows that the transfer reaches two goals we set: the transfer maintains the label, yet still allows a large degree of flexibility (on par with reconstruction in terms of diversity, which is the upper threshold) within classes to transfer the local changes.

Transfer	MNIST $\rightarrow$ MNIST	MNIST $\rightarrow$ Fashion MNIST	Fashion MNIST $\rightarrow$ MNIST	MNIST $\rightarrow$ SC09	SC09 $\rightarrow$ MNIST
Accuracy	0.98	0.945	0.890	0.670	0.982

Table 2: Domain Transfer Accuracy for MNIST $\leftrightarrow$ MNIST , MNIST $\leftrightarrow$ Fashion MNIST and MNIST $\leftrightarrow$ SC09 transfer respectively.

3.3 Interpolation

We use interpolation as a proxy to visualization the behavior of latent space under transferring. Particularly, we are interested in two comparing three rows of interpolations: (1) the interpolation in the source domain's latent space, which shows the property of pre-trained generative model and can serve as a baseline, (2) transfer fixed points of interpolation to the target domain's latent space and interpolate in that space, and (3) transfer all points of interpolation to the target domain's latent

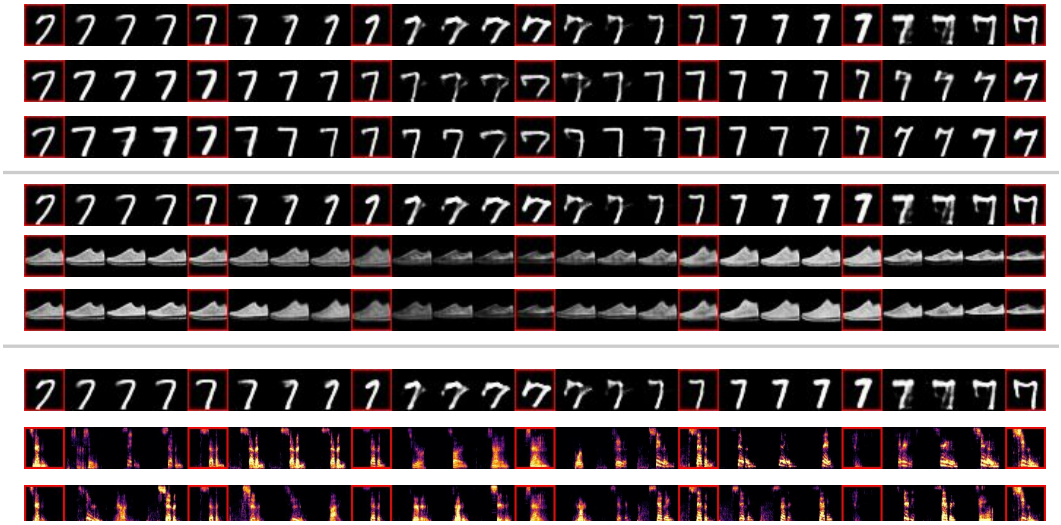

Figure 5: Intra-class Interpolation. Interpolations are divided into three groups, representing MNIST $\rightarrow$ MNIST , MNIST $\rightarrow$ Fashion MNIST and MNIST $\rightarrow$ SC09 transfer respectively from top to bottom. Images in a red square are fixed points in the interpolations, which means interpolation happens between two neighboring fixed points. In each group, there are three rows of interpolations: (1) Interpolate in source domain between fixed data points, (2) Transfer fixed data points in source domain to target domain and interpolate between transferred fixed points there, (3) Transfer all points in first row to the target domain. It can be shown that in this intra-class setting, transferring preserves the smoothness of data when interpolating within one class.

Figure 6: Inter-class Interpolation. The arrangement of images is the same as Figure 5, except that interpolation now happens between classes. It can be shown that, unlike regular generative model (row 1 and row 2 in each group) that exhibits pixel (data) level interpolation, especially the bluriness and distortion half way between instances of different labels, our proposed transfer (row 3) resorts to produce high-quality, in-domain data. This is an expected behavior since our proposed method learns to model the marginalized posterior of data distribution.

space, which shows how the transferring warps the latent space. We use spherical interpolation (e.g., $\sqrt{p}v_1 + \sqrt{(1-p)}v_2$) [3] since we are interpolating in the Gaussian latent space. We show inter-class and inter-class interpolation in Figure 5 and Figure 6 respectively.

[3] https://www.inference.vc/high-dimensional-gaussian-distributions-are-soap-bubble/

3.4 ANALYSIS

Since our method is a semi-supervised method, We want to know how effectively our method leverages the labeled data. In Table 3 we show for the MNIST $\rightarrow$ MNIST setting the performance measured by transfer accuracy with respect to the number of labeled data. This shows that our method could effectively leverage extra labeled data when it becomes available.

# Supervised	0	10	100	1000	10000	60000
Accuracy	0.1390	0.339	0.524	0.6810	0.898	0.980

Table 3: Data Efficiency. The number of supervised data points is for all 10 digits.

Furthermore, to investigate the contribution from our proposed components, we explore the scenario where VAE is unconditional or the SWD is not employed in Table 4. It can be shown that these proposed components doe bring a huge boost in performance.

Data Domain	Vanilla, Unconditional VAE	Conditional VAE	Conditional VAE + SWD
Accuracy	0.149	0.849	0.980

Table 4: Component Contribution

The training time for the conditional VAE on shared space is significantly shorter than that of preparing pre-trained model. For example, for MNIST $\leftrightarrow$ MNIST our methods produce high-quality results within several minutes of training while pre-trained model takes hours. The comparison is more observable for SC09's case where our method takes half an hour while the pre-trained model takes days to train.

4 RELATED WORK

We discuss three lines of researches that are related to our work.

Generative Model: Generative models with latent space are usually done in a fashion that transfer a simple, tractable distribution $p(z)$ into the approximation of population distribution $p^*(x)$, usually completed through the use of deep neural network based. Such models include VAE (Kingma & Welling, 2013) and GAN (Goodfellow et al., 2014). GANs are trained with an accompany classifier that attempts to distinguish between samples from the decoder and the true dataset. VAEs, in the contrast, are trained with an encoder distribution $q(z|x)$ as an approximation to the posterior $p(z|x)$ using variational approximation through the use of evidence lower bound (ELBO). Many studies have been focusing since then, including conditional generation (Mirza & Osindero, 2014), Generation of one domain conditioned on another one (Dai et al., 2017; Reed et al., 2016), generation of high-quality image (Karras et al., 2018) and music (Roberts et al., 2018)

Latent Space: The choose of latent space for training generative model has been studied intensively to improve the quality, diversity, or the training stability of generative model (Gulrajani et al., 2017; Li et al., 2017; Bikowski et al., 2018). Recently it is proposed that post-hocing the latent space of pre-trained generative model could control its behavior in a desired direction (Engel et al., 2018)

Domain Transfer: The domain transfer enables transfer between images (Isola et al., 2016; Zhu et al., 2017; Li et al., 2018; Li, 2018), audio (Mor et al., 2018) and video (Wang et al., 2018), intuitively mapping between two domains where the variations in one domain should be reflected in the other. Besides visually appealing results, domain transfer also enables application such as image colorization Zhang et al. (2017). Domain transfer is also porpoised to be done through jointly training of generative models (Lu, 2018). Also, the behavior of domain transfer models also attracts attention. For example, Chu et al. (2017) suggests that image transfer does only local, texture level transfer.

REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Mikoaj Bikowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=r1lUOzWCW`.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015.

Casey Chu, Andrey Zhmoginov, and Mark Sandler. Cyclegan: a master of steganography. *arXiv preprint arXiv:1712.02950*, 2017.

Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. Towards diverse and natural image descriptions via a conditional gan. *arXiv preprint arXiv:1703.06029*, 2017.

Ishan Deshpande, Ziyu Zhang, and Alexander Schwing. Generative modeling using the sliced wasserstein distance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3483–3491, 2018.

Chris Donahue, Julian McAuley, and Miller Puckette. Synthesizing audio with generative adversarial networks. *arXiv preprint arXiv:1802.04208*, 2018.

Jesse Engel, Matthew Hoffman, and Adam Roberts. Latent constraints: Learning to generate conditionally from unconditional generative models. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=Sy8XvGb0-`.

Patrick Feaster. The phonautographic manuscripts of édouard-léon scott de martinville. *FirstSounds.org*, 2009.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5767–5777. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.pdf`.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arxiv*, 2016.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*, pp. 2203–2213, 2017.

Jerry Li. Twin-gan–unpaired cross-domain image translation with weight-sharing gans. *arXiv preprint arXiv:1809.00946*, 2018.

Minjun Li, Haozhi Huang, Lin Ma, Wei Liu, Tong Zhang, and Yu-Gang Jiang. Unsupervised image-to-image translation with stacked cycle-consistent adversarial networks. *arXiv preprint arXiv:1807.08536*, 2018.

Yingjing Lu. Cross domain image generation through latent space exploration with adversarial loss. *arXiv preprint arXiv:1805.10130*, 2018.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Noam Mor, Lior Wolf, Adam Polyak, and Yaniv Taigman. A universal music translation network. *arXiv preprint arXiv:1805.07848*, 2018.

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical latent vector model for learning long-term structure in music. *arXiv preprint arXiv:1803.05428*, 2018.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*, 2018.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999*, 2017.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

APPENDIX A TRAINING TARGET DESIGN

We want to archive following three goals for the proposed VAE for latent spaces:

1. It should be able to model the latent space of both domains, including modeling local changes as well.
2. It should encode two latent spaces in a way to enable domain transferability. This means encoded z_1 and z_2 in the shared latent space should occupy overlapped spaces.
3. The transfer should be kept in the same class. That means, regardless of domain, zs for the same class should occupy overlapped spaces.

With these goals in mind, we propose to use an optimization target composing of three kinds of losses. In the following text for notational convenience, we denote approximated posterior $Z'_d \triangleq q(z'|z_d, D = d), z_d \sim q(z_d|x_d), x_d \sim p(x_d)$ for $d \in \{1, 2\}$, the process of sampling z'_d from domain d.

1. Modeling two latent spaces with local changes. VAEs are often used to model data with local changes in mind, usually demonstrated with smooth interpolation, and we believe this property also applies when modeling the latent space of data. Consider for each domain $d \in \{1, 2\}$, the VAE is fit to data to maximize the ELBO (Evidence Lower Bound)

$$\mathcal{L}_d^{\text{ELBO}} = \mathbb{E}_{z' \sim Z'_d} \left[\log \pi(z_d; g(z', D = d)\right] - \beta_{\text{KL}} D_{\text{KL}} \left(q(z'|z_d, D = d) \| p(z')\right)$$

where both q and g are fit to maximize $\mathcal{L}_d^{\text{ELBO}}$. Notably, the latent space zs are continuous, so we choose the likelihood $\pi(z; g)$ be the product of $\mathcal{N}(z; g, \sigma^2 I)$, where we set σ to be a constant that effectively sets $\log \pi(z; g) = ||z - g||_2$, which is the L2 loss in Figure 1 (a). Also, D_{KL} is denoted as KL loss in Figure 1 (a).

2. Cross-domain overlapping in shared latent space. Formally, we propose to measure the cross-domain overlapping through the distance between following two distributions as a proxy: the distribution of z' from source domain (e.g., $z'_1 \sim Z'_1$) and that from the target domain (e.g., $z'_2 \sim Z'_1$). We use Wasserstein Distance (Arjovsky et al., 2017) to measure the distance of two sets of samples (this notion straightforwardly applies to the mini-batch setting) S'_1 and S'_2, where S'_1 is sampled from the source domain $z'_1 \sim Z'_1$ and S'_1 from the target domain $z'_2 \sim Z'_d$. For computational efficiency and inspired by Deshpande et al. (2018), we use SWD, or Sliced Wasserstein Distance (Bonneel et al., 2015) between S'_1 and S'_2 as a loss term to encourage cross-domain overlapping in shared latent space. This means in practice we introduce the loss term

$$\mathcal{L}^{\text{SWD}} = \frac{1}{|\Omega|} \sum_{\omega \in \Omega} W_2^2 \left(\text{proj}(S'_1, \omega), \text{proj}(S'_2, \omega)\right)$$

where Ω is a set of random unit vectors, $\text{proj}(A, a)$ is the projection of A on vector a, and $W_2^2(A, B)$ is the quadratic Wasserstein distance, which in the one-dimensional case can be easily solved by monotonically pairing points in A and B, as proven in Deshpande et al. (2018).

3. Intra-class overlapping in shared latent space. We want that regardless of domain, zs for the same class should occupy overlapped spaces, so that instance of a particular class should retain its label through the transferring. We therefore introduce the following loss term for both domain $d \in \{1, 2\}$

$$\mathcal{L}_d^{\text{Cls}} = \mathbb{E}_{z' \in Z'_d} H(f(z'), l_{x'})$$

where H is the cross entropy loss, $f(z')$ is a one-layer linear classifier, and $l_{x'}$ is the one-hot representation of label of x' where x' is the data associated with z'. We intentionally make classifier f as simple as possible in order to encourage more capacity in the VAE instead of the classifier. Notably, unlike previous two categories of losses that are unsupervised, this loss requires labels and is thus supervised.

In Figure 7 we show the intuition to design and the contribution to performance from each loss terms.

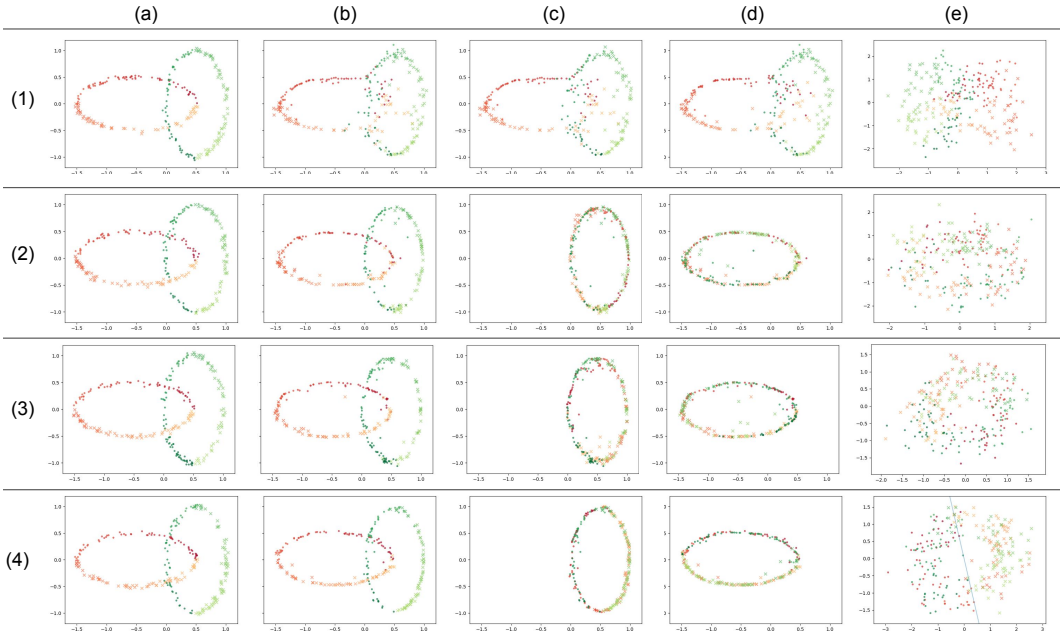

Figure 7: Synthetic data to demonstrate the transfer between 2-D latent spaces with 2-D shared latent space. Better viewed with color and magnifier. Columns (a) - (e) are synthetic data in latent space, reconstructed latent space points using VAE, domain 1 transferred to domain 2, domain 2 transferred to domain 1, shared latent space, respectively, follow the same arrangement as Figure 2. Each row represent a combination of our proposed components as follows: **(1)** Regular, unconditional VAE. Here transfer fails and the shared latent space are divided into region for two domains. **(2)** Conditional VAE. Here exists an overlapped shared latent space. However the shared latent space are not mixed well. **(3)** Conditional VAE + SWD. Here the shared latent space are well mixed, preserving the local changes across domain transfer. **(4)** Conditional + SWD + Classification. This is the best scenario that enables both domain transfer and class preservation as well as local changes. It is also highlighted in Figure 2. An overall observation is that each proposed component contributes positively to the performance in this synthetic data, which serves as a motivation for our decision to include all of them.

APPENDIX B MODEL ARCHITECTURE

The model architecture of our proposed VAE is illustrated in Figure B. The model relies on Gated Mixing Layers, or GML. We find empirically that GML improves performance by a large margin than linear layers, for which we hypothesis that this is because both the latent space (z_1, z_2) and the shared latent space z' are Gaussian space, GML helps optimization by starting with a good initialization. We also explore other popular network components such as residual network and batch normalization, but find that they are not providing performance improvements. Also, the condition is fed to encoder and decoder as a 2-length one hot vector indicating one of two domains.

For all settings, we use dimension of shared latent space 100, $\beta_{\text{SWD}} = 1.0$ and $\beta_{\text{CLs}} = 0.05$,

Specifically, for MNIST $\leftrightarrow$ MNIST and MNIST $\leftrightarrow$ Fashion MNIST, we use dimension of shared latent space 8, 4 layers of FC (fully connected layers) of size 512 with ReLU, $\beta_{\text{KL}} = 0.05$, $\beta_{\text{SWD}} = 1.0$ and $\beta_{\text{CLs}} = 0.05$; while for MNIST $\leftrightarrow$ SC09, we use dimension of shared latent space 16, 8 layers of FC (fully connected layers) of size 1024 with ReLU $\beta_{\text{KL}} = 0.01$, $\beta_{\text{SWD}} = 3.0$ and $\beta_{\text{CLs}} = 0.3$. The difference is due to that GAN does not provide posterior, so the latent space points estimated by classifier is much harder to model.

For optimization, we use Adam optimizer(Kingma & Ba, 2014) with learning rate 0.001, $beta_1 = 0.9$ and $beta_2 = 0.999$. We train 50000 batches with batch size 128. We do not employ any other tricks for VAE training.

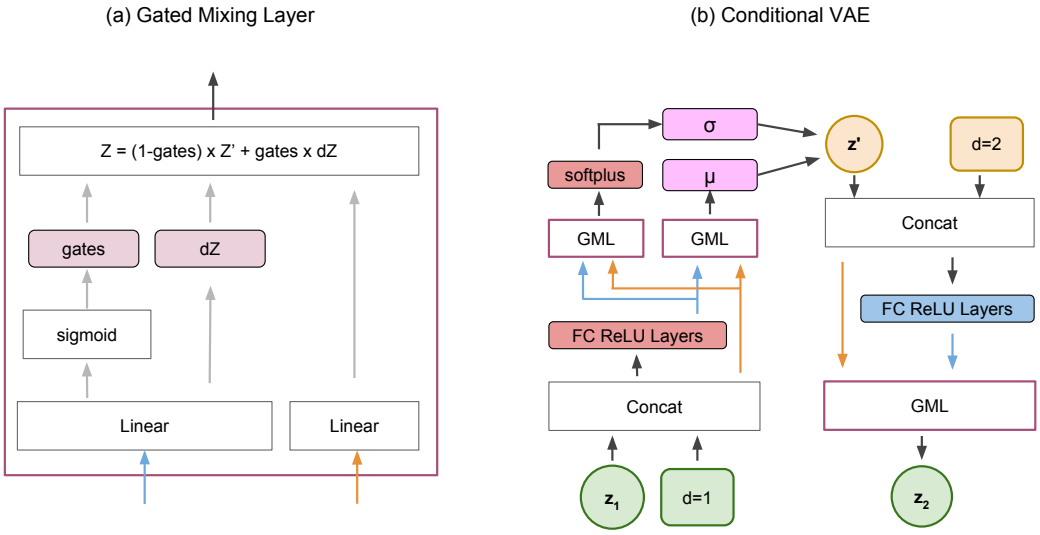

Figure 8: Model Architecture for our Conditional VAE. **(a)** Gated Mixing Layer, or GML, as an important building component. **(b)** Our conditional VAE with GML.

APPENDIX C SUPPLEMENTARY FIGURES

MNIST Digits	Fashion MNIST Class
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

Table 5: MNIST Digits to Fashion MNIST class Mapping. This mapping is made according to Labels information available at `https://github.com/zalandoresearch/fashion-mnist`