A Fully-Unsupervised Generative Method for Choreo-Musical Translation

Sam Lowe
SUNet ID: samlowe

1 Abstract

Creativity is a core component of human intelligence. It allows us to adapt to problems as they arise and transfer knowledge from old domains to new ones. Clearly, this type of creativity would be advantageous if it could be imbued in our learning algorithms, as it would enable us to build on gains from one task by applying them to another, rather than training agents from scratch for each new task. In order to promote this type of transfer learning, our algorithms need a method of unifying data representations across domains and performing domain-to-domain translation. In this paper, I introduce a fully unsupervised method for performing domain transfer via a variational autoencoder bridging model, an application of the method to the problem of bi-directional music to dance translation, and a dataset of over 1300 paired samples of music and choreography sequences. The method is able to achieve low reconstruction accuracy even when passed through the bridging model, but the diversity of generated samples is ultimately limited by the size of the dataset it was trained on.

2 Introduction

One of the hallmarks of "Strong AI" is the ability to rapidly adapt to changing environments and problem settings, using past lessons to inform new solutions in *creative* ways. This ultimate aim is one of the core reasons I am fascinated by applications of AI to creative domains, as creativity is a fundamental tenet of intelligence in a broad sense. One of the many tasks that is often involved in the creative process is the act of translation from one domain to another. A prime example of this is the process by which dancers translate incoming musical cues as motion. Another could be a landscape painter who translates incoming visual, perceptual phenomena into a hierarchical model of light and color and ultimately into a series of fine motor movements that will reproduce that phenomenon on paper. It is clear that AI systems applied to these tasks need similar translation skills, but the ability to translate across domains could be a crucial step towards strong AI more generally, as it can enable an agent to learn a unified representation of the world across modalities.

In this project, I will focus on the translation of music to dance and vice versa. Such a system could be applied to a myriad of performance and educational settings, like choreography with dancer-as-composer, integrated collaborative pieces, and low-barrier social music making, and should provide a solid test bed for this type of machine translation as both source signals are very high dimensional but do have a fundamental relationship to one another. Our algorithm will be able to take as input either music (represented as MIDI) or dance (represented as a sequence of poses) and output a semantically meaningful segment of the opposite domain that represents choreography and music that should be paired together. This will be accomplished via a series of variational autoencoders that encode the original inputs and then translate between the embeddings with a purpose-built bridging VAE.

CS229: Machine Learning, Autumn 2021-2022

3 Related Work

Several projects have studied the problem of unidirectional music-to-dance translation. In their GrooveNet paper [1], Alemi et al. proposed a model for generating dance from music using an architecture consisting of Factored Conditional Restricted Boltzmann Machines and Recurrent Neural Networks, though they struggled to generalize beyond examples in the training set. Music2Dance [2] is another proposed architecture for dance generation that uses Convolution Neural Networks, LSTM layers, and Mixture Density Networks to process incoming audio into musical features that are used as input to the dance-generation network. While both models show potential for realistic results, neither is capable of the bi-directional translations that I aim to achieve.

The current state-of-the-art for domain translation leverages models that learn latent representations of the original data to perform translation in the lower-dimensional latent space. Tian & Engel [3] proposed a method for performing such latent translation using pretrained VAEs and paired data across the visual and audio domains to train a model to translate between written and spoken numbers. The bridging VAE that they propose is optimized using a three-termed loss function consisting of the standard VAE EBLO, a sliced Wasserstein distance term to encourage paired data to have similar embeddings, and a classification term that enforces semantic structure on the latent space based on the original data labels. This approach allows for the types of bi-directional translation that I am interested in, and will form the core inspiration for my project methodology. However, in comparison to their method, which requires label supervision, my bridging model will be trained fully unsupervised and include only the ELBO and sliced Wasserstein distance terms.

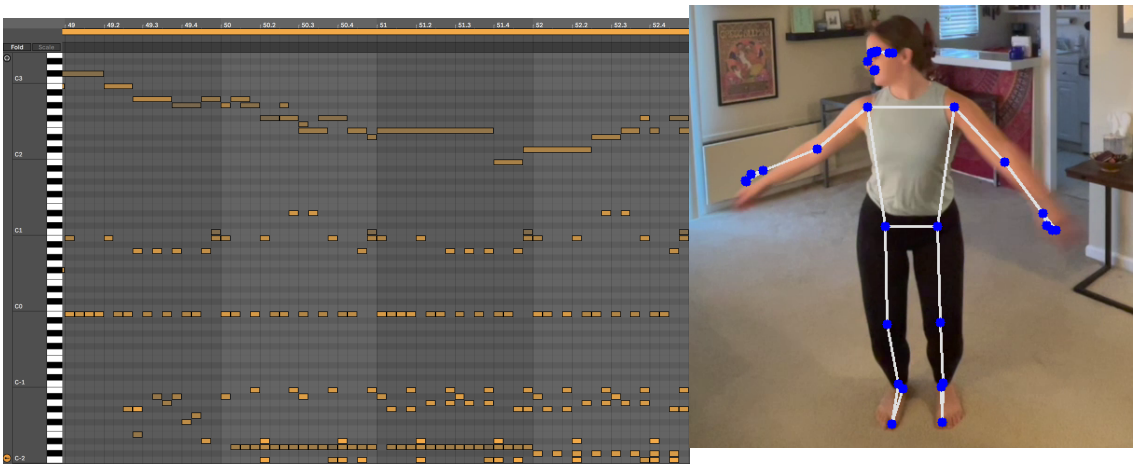
Given that the uni-modal components of the translation model are themselves variational autoencoders, I have also explored research that applies VAEs to music or dance in isolation to better understand what sort of architectures will be the most amenable to the problem setting. Augello et al. [4] implemented a standard VAE model to generate choreography for a humanoid robot. In comparison, Nagano et al. [5] built a more custom-purpose VAE model inspired by HMMs and hierarchical Dirichlet processes to encourage the model to learn contiguous embeddings that could be used for downstream motion segmentation tasks. For music, much of the work on VAEs has been dominated by Magenta's MusicVAE model [6], which uses Recurrent Neural Networks to augment the ability of the core VAE to produce full sequences of melodies. Across the VAE and non-VAE approaches to music and dance generation, most models relied on LSTM or other RNN-style layers to model long-term sequence dependencies, but frequently report problems with prior collapse because of the power of the autoregressive models. Our method attempts to circumvent this failure point by training on, and thus decoding, the full sequence directly, rather than seeding an autoregressive generator.

4 Dataset & Features

The dataset for this project was constructed from scratch because I needed fully-paired dance and music data, and I was also interested in testing the ability of the models to adapt to the musical and choreographic tastes of particular individuals. Creating the dataset was by far the biggest time investment for this work, as composing a sufficient number of samples of original music with the needed level of variety required many hours of effort. I ultimately ended with 1329 measures of music consisting of drum, bass, and melody parts stacked in a single MIDI message (the drums ranging from C-2 to C-1, bass from C-1 to C1, and melody above C1). The MIDI was quantized to sixteenth note steps and then converted into a (16*1329)-by-128 matrix (which I'll call M) where $M_{ij} = 1$ if MIDI note j was sounding at step i. I then ran a 2-bar (32 step) sliding window over the data with a stride of 1 and flattened each 2-bar segment, resulting in 1328 samples of dimensionality $32 * 128 = 4096$.

After the music was composed, I collected video data of a dancer improvising to the full 1329-bar piece. I then quantized the frames at a rate synchronous to the sixteenth notes in the music, and passed the resulting frames through PoseNet to recover 33, 3-dimensional joint positions at each step. The sequence of poses was similarly passed through a 2-bar sliding window, resulting in 1328 samples of dimensionality $32 * 33 * 3 = 3168$. A sample of the raw pose and MIDI data is provided in Figure 1.

After the above-described processing steps, I elected not to perform any further data cleaning, such as normalization, as the data from both domains was already sufficiently well-constrained - the MIDI



(a) Sample unified MIDI data

(b) Sample pose frame

Figure 1: Data samples from each domain

data was binary encoded and the joint positions ranged from -1 to 1. The samples for both domains were split into training, validation, and testing sets of size 1128, 100, and 100 respectively.

5 Method

As mentioned above the core of my methodology is an adapation of Tian & Engel's Latent Translation [3] approach to a fully unsupervised domain. The method consists of a trio of variational autoencoders: one each to encode the dance and music data into latent spaces of equivalent dimension, and a third to take in those initial embeddings and learn a latent space where paired data develops significant overlapping to enable translation between the domains. In order to more fully specify the methodology, I will begin with a brief treatment of variational autoencoders in general, then describe the particularities of the bridging VAE, and finally the optimization method used in training (Adam).

5.1 VAEs

Variational autoencoders are a type of generative model that attempt to learn a latent space representing a compressed version of the original data, with restrictions on the structure of the latent space that are aimed at enabling the generation of new data points by sampling the latent distribution. The modeling of these distributions consists of three parts: an encoder network that models $p(z|x)$ to recover the latent codes for a data point, a decoder network that models $p(x|z)$ to generate the data from the latent codes, and an assumption that $p(z)$ is Gaussian. Variational autoencoders are trained by minimizing a loss function termed the Evidence Lower Bound (ELBO), which has the form:

$$\mathcal{L}^{ELBO} = -\mathbb{E}_{z\sim Q}[logp(x|z;\theta)] + \beta_{KL} D_{KL}(Q||p_z)$$

Where Q is the assumed Gaussian distribution over z and θ are our model parameters. The ELBO loss can be considered a two-termed loss function where the first term measures the quality of the reconstructions and the second the strength of our regularization assumption over p_z, with β_{KL} as a hyper parameter that controls the trade off between the two.

5.2 Bridging VAE

The bridging VAE for my method is a simple extension of the basic VAE construction described above with a slightly different loss function. Because I want to be able to translate between data domains, the bridging VAE has the express goal of embedding the latent codes of paired data points to be very near to each other in its latent space. To capture this desire, the ELBO is extended to

include a Sliced Wasserstein Distance term to capture the distribution distance between the bridged embeddings of paired data minibatches (z'_1, z'_2):

$$\mathcal{L}^{SWD} = \frac{1}{|\Omega|} \sum_{w \in \Omega} W^2(proj(z'_1, w), proj(z'_2, w))$$

Where Ω is a set of random unit vectors, $proj(I, j)$ is the projection of I onto j, and W^2 is the squared Wasserstein distance. Adding this term to the ELBO loss above gives us a full loss function:

$$\mathcal{L} = \mathcal{L}^{ELBO} + \beta_{SWD}\mathcal{L}^{SWD}$$

Where β_{SWD} is again a hyperparameter specifying the relative weighting.

5.3 Optimization

For all three VAE models, the optimization method utilized was Adam. Adam is an optimization method that computes adaptive learning rates based on de-biased estimates of the first and second moments of the gradients, $\hat{m}_t$ and $\hat{v}_t$. The update rule then has the form:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t$$

Where η is the base learning rate and ϵ is a small positive constant to prevent dividing by zero. Adam's adaptive learning rates are less susceptible to local deviations in the loss landscape than SGD, and it has been observed to be highly performant for a wide variety of applications, making it the default choice for optimization across much of the literature

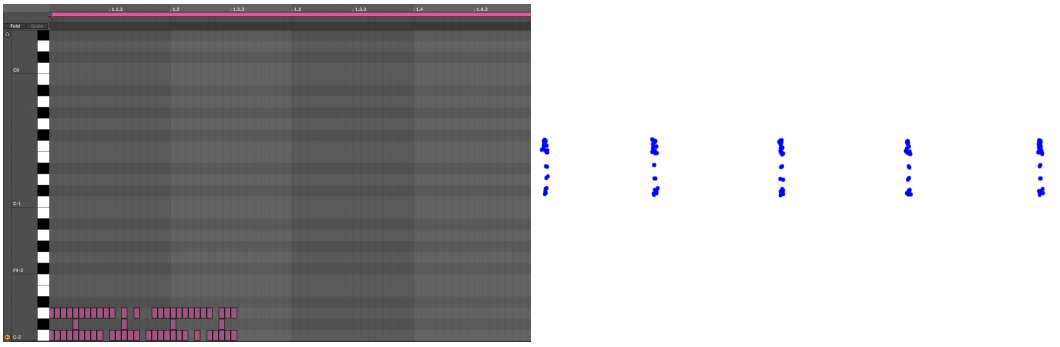
6 Experimental Results & Discussion

My first set of experiments consisted of a hyperparameter grid search over all three models, using the reconstruction error on the held-out validation set as the metric for comparison. The hyperparameters that were searched over included layer number and sizes, latent (output) dimension, learning rates, KL divergence weights, and dropout rates. Each model was trained for a total of 50 epochs. The MIDI and pose data VAEs had the same optimality point consisting of three linear layers sized 512, 256, and 128, a latent dimension of 128, a learning rate of 0.001, a KL divergence weight of 0.1, and no dropout. The encoder and decoder structures were mirrored, so the decoder architecture is three layers of size 128, 256, and 512. The MIDI VAE had one key difference in that its final decoder layer included a TanH activation, since our original data was in the [0, 1] range. For the bridging VAE, the optimal architecture was two layers of size 256 and 128, a latent dimension of 32, a learning rate of 0.001, a KL divergence weight of 0.005, and no dropout. The sliced Wasserstein distance weighting was set at 10 in all trials to scale it similarly to the ELBO term.

Using the architectures described above, I then performed a variety of experiments to measure the reconstruction and generation ability of the network trio. For reconstruction accuracy, the metric used was the mean squared error between the original and reconstructed data. The two original VAEs were tested for reconstruction performance on the test set for their data domain, while the bridging VAE was measured by music-to-music reconstruction, dance-to-dance reconstruction, embedding-to-embedding reconstruction, music-to-dance reconstruction, and dance-to-music reconstruction. These results can be found in Table 1.

Model	MIDI VAE	Pose VAE	Bridge (Music-Music)
MSE	0.025	0.005	0.025
Bridge (Dance-Dance)	Bridge (Embed-Embed)	Bridge (Music-Dance)	Bridge (Dance-Music)
0.005	2.006	0.005	0.025

Table 1: Reconstruction results by mean-squared error on held out test set.



(a) Sample generated MIDI frame

(b) Sample generated pose sequence

Figure 2: Generated samples from each domain

Finally, I explored the ability of the full pipeline to generate novel data by observing a variety of samples from the bridging model decoded in both domains, an example of which is given in Figure 2.

Based on the quantitative results, the models were all able to achieve a relatively low reconstruction error, a phenomenon that was preserved even when translating across the bridging VAE, indicating that the bridging VAE is learning adequately. While the embedding-to-embedding loss appears quite high in comparison, we don't know the exact scaling of the latent codes from either of the MIDI VAE or Pose VAE models, so it is comparatively less important of a metric.

Closer examination of the generated samples paint a less optimistic picture of the model's performance. The main conclusion I drew from observing the results were that the dataset was likely not sufficiently sized to enable proper learning, as the models appear to have a high degree of uncertainly. More specifically, when observing the dance data, the range of motion is highly constrained, and when generating the MIDI data, it was necessary to consider any prediction higher than 0.25 a note-on event in order to get any output at all. The model seems to be making these low-magnitude predictions to guard against errors, indicating that the data was not enough to teach it to maximize performance rather than mitigate errors. However, that is not to say that there are no positive takeaways from the generated samples, as the model is certainly learning some semantically meaningful lessons from the data. For example, in the generated pose data, the model appears to have a consistent understanding of human proportions and the distribution of the joint positions around the human form, and the MIDI data does show some initial understanding of rhythm as it is able to generate a drum pattern in the lower part of the register. The second of these observations further supports the notion that the training regime was too data-poor, as a large number of samples had this sort of straight ahead drum, hi-hat, and snare rhythm, so the model was able to capture it due to the comparatively higher number of examples. Based on these observations, I do not think that the models overfit during training - if anything, they underfit.

7 Conclusion

In this work, I introduced a fully-unsupervised, end-to-end method for domain translation based on variational autoencoders and explored its potential application to the task of bi-directional music and dance translation. The approach shows promise in its ability to achieve a low reconstruction error and model semantically-meaningful components of the underlying data distribution, but was ultimately limited by the size of the data set I was able to construct. Future work on this problem would center around acquiring a larger dataset to see if we can extract even deeper semantic structure from the domains of interest, either by expanding the dataset I already have or by collecting choreography data for non-original music (like the Million Songs Dataset).

Contributions

Many thanks Marissa Kuczkowski for her contributions to the dataset as the primary dancer. Some of the code for the VAE models and SWD metric was adapted from https://github.com/AntixK/PyTorch-VAE and https://github.com/eifuentes/swae-pytorch.

References

[1] Alemi, O., Francoise, J., & Pasquier, P. (2017) GrooveNet: Real-Time Music-Driven Dance Movement Generation using Artificial Neural Networks. In *Workshop on Machine Creativity at ACM SIGKDD.*

[2] Zhuang, W., Wang, C., Xia, S., Chai, J., & Wang, W. (2020) Music2Dance: DanceNet for Music-driven Dance Generation. In *arXiv e-prints*.

[3] Tian, Y., & Engel, J. (2019) Latent Translation: Crossing Modalities by Bridging Generative Models. In *arXiv e-prints*.

[4] Augello, A., et al. (2017) Creative Robot Dance with Variational Encoder. In *arXiv e-prints*.

[5] Nagano, M., et al. (2019) High-dimensional Motion Segmentation by Variational Autoencoder and Gaussian Processes. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[6] Roberts, A., et al. (2019) A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music. In *arXiv e-prints*.