DISENTANGLING MOTION FROM IDENTITY WITH BARLOW TWINS AUTOENCODERS

CMPT 732 (FALL 2021) TERM PROJECT REPORT

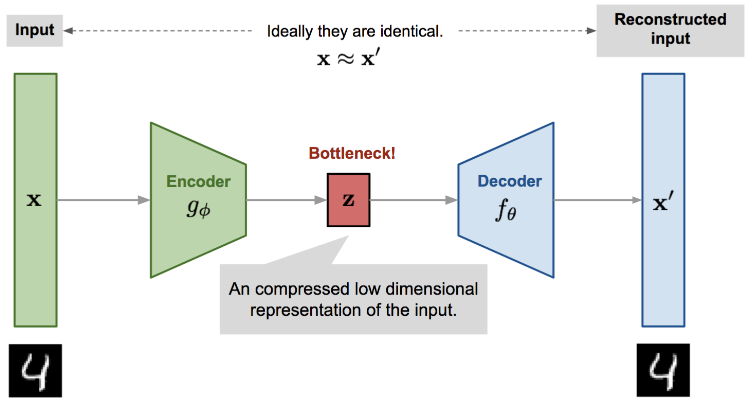
Geoffrey Mantel: 893015835

ABSTRACT

This paper attempts to disentangle motion from identity in the latent space of an autoencoder trained on adjacent frames of the "MovingMNIST" video dataset. The intuition is that the embeddings of the identity of objects in a video (the "what" embeddings) should be invariant between two adjacent frames of the same video, whereas the embeddings of the motion of those objects (the "where" embeddings) can vary freely with time. This approach is inspired by the brain's separate "what" versus "where" visual processing pathways.

1 Introduction

Autoencoders are a type of neural network that learn a compressed representation of their input by funnelling the high-dimensional input through a lower-dimensional pinch-point. The goal of an autoencoder is to learn a compressed representation of the input such that the input can be reproduced from only this information. An encoder learns this representation, called the "latent vector", and a decoder learns how to regenerate the original input from the latent vector. Restricting the dimensionality of the latent space forces the network to learn the important spatial details of the input image.



Input
Ideally they are identical.
$\mathbf{x} \approx \mathbf{x}'$
Reconstructed input
$\mathbf{x}$
Encoder
g_ϕ
Bottleneck!
$\mathbf{z}$
Decoder
f_θ
$\mathbf{x}'$
An compressed low dimensional representation of the input.

Figure 1: Autoencoder [Weng [2018]]

In terms of video frame prediction and generation, autoencoders represent a very appealing base architecture, since video datasets are often lacking labels, and autoencoders can be trained without supervision. Many techniques exist to cope with video motion in an autoencoder: enhancements to the network structure, augmentation of the training data, incorporation of real-world physics into the loss function, etc. [Oprea et al. [2020])]

1.1 What versus Where

It has been proposed [Goodale and Milner [1992]] that the human visual system consists of two distinct processing streams: a "what" pathway, which performs object recognition, and a "where" pathway, which processes the spatial properties of the objects in a scene. These two different systems have different properties of processing speeds and internal representations of visual content.

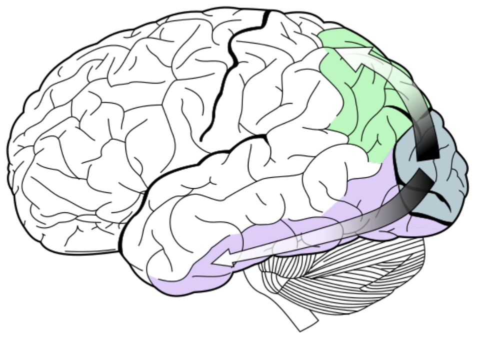


Figure 2: "What" (lower) versus "Where" (upper) Streams

Taking inspiration from this structure, this paper will present a video autoencoder architecture consisting of two separate encoders, which concatenate their output into a single decoder. The goal of this architecture is to disentangle the motion information from the identity of the contents of the video frames - to separate the "what" from the "where". Intuitively, the "what" encoding of two consecutive video frames should be as similar as possible, whereas the "where" encoding could change. This intuition will be explored through a variety of loss terms.

It should be noted that the terminology of "what" and "where" imply semantics which are more specific than what is being attempted in this paper. This paper does not classify objects, nor does it locate them in space. Instead, it simply attempts to process a consecutive sequence of video frames to determine if they can be split into a constant term which describes the contents of the video as a whole, and a varying term which transforms the constant into a time-varying, frame-specific vector.

In mathematical terms, if we have an encoder $g(x)$ which maps image pixels x onto a latent vector $z \in \mathbb{R}^d$, then a video of n frames of the same objects moving in space should be decomposable into a constant z "identity" (or "what") vector plus a changing z' "motion" (or "where") vector:

$$\{z_1, z_2, \dots z_n\} = \{z + z'_1, z + z'_2, \dots z'_n\} \tag{1}$$

Succeeding in this effort would have implications for object classification and future-frame prediction in videos, in that disentangled "what" vs "where" representations should be easier to learn.

2 Experimental Setup

2.1 Moving MNIST Dataset

This paper will base its experiments on the "MovingMNIST" [Srivastava et al. [2015]] dataset. MovingMNIST is a dataset consisting of 10k sequences of 20 frames showing 2 hand-written digits moving in a 64x64 frame. The digits themselves are taken from the popular MNIST dataset [Deng [2012]]. The digits move along linear trajectories, and bounce off the edges of the frame in predictable fashion. No information about these trajectories will be used in the training of our autoencoder - the goal is to disentange identity from motion in a generic fashion, with no prior on the motion or contents of the videos.

2.2 Model Architecture

The general model architecture for this project consists of two encoders and a decoder. We will denote the "what" encoder as $g(x) = z$, the "where" encoder as $g'(x) = z'$, and the decoder as $f(z, z') = \hat{x}$.

The "what" encoder will be fed two consecutive frames of video, reusing the same weights for each frame. The "where" encoder will be fed the first frame of video. The output of each encoder is concatenated together before being sent as input into the decoder, such that the dimensionality of the decoder's input vector is the sum of the dimensionality of the vectors of the two encoders.

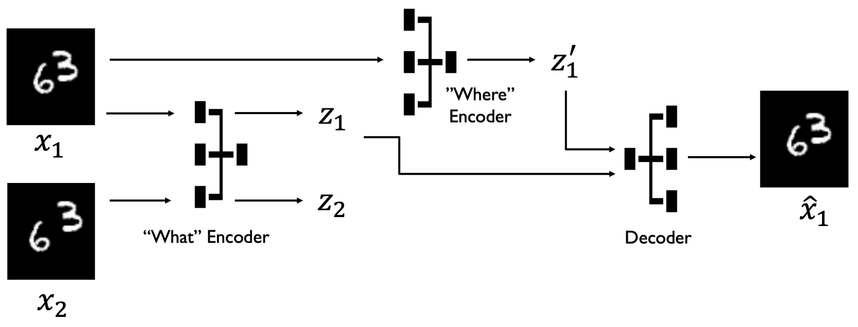


Figure 3: Model Architecture

Both encoders share the same low-level structure: 3 blocks, each consisting of a 2-dimensional convolution layer with ReLU activation, batch normalization, and max pooling. The final layer is a Linear (fully-connected) layer of dimension d (where the encoder outputs $z \in \mathbb{R}^d$), and a 1-dimensional batch normalization layer (discussed in the "results" section). The dimensionality of the latent vector is chosen to be different between the "what" and the "where" encoder, with higher dimensionality on the "what" encoder. This design decision was an attempt to prevent the "where" encoder from learning too much of the identity information.

The decoder is essentially the inverse of the encoders: 3 blocks of deconvolution operations with batch normalization and upsampling. The final layer of the decoder has a sigmoid activation, to encourage the output to conform to the greyscale $[0, 1]$ pixel value range of the MNIST dataset.

In all of the experiments, the Pytorch Adam optimizer was used, with a learning rate of 0.001. Pytorch's SGD optimizer was also tried, but Adam consistently gave better results.

3 Results

3.1 Inertial Loss Function

This project's first attempt to disentangle identity from motion involved minimizing the mean-squared error between the embedded vector of two adjacent frames of video:

$$L_{\text{I}} = \frac{1}{n}\sum_{n} ||z_2 - z_1||_2^2 \tag{2}$$

As mentioned earlier, the last layer of the encoder network is a 1-dimensional batch normalization layer, which discourages the encoder from minimizing L by outputting the zero vector.

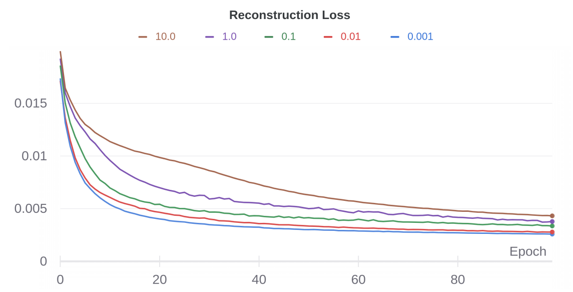
This loss term is combined with the usual autoencoder reconstruction loss term measuring the mean-squared error in the pixel values between the input x and the decoder output $\hat{x}$:

$$L_{\text{R}} = \frac{1}{n}\sum_{n} ||\hat{x}_1 - x||_2^2 \tag{3}$$

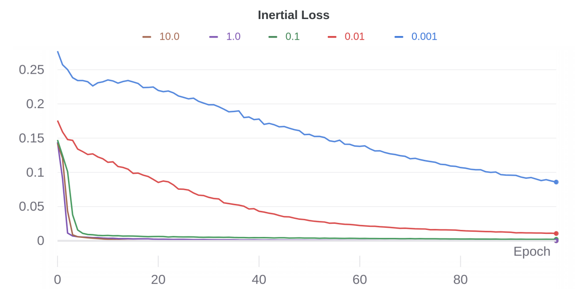
These two loss terms are combined, with a constant mixing term λ, to produce the total loss term use to train the network:

$$L_{\text{Total}} = L_{\text{R}} + \lambda L_{\text{I}} \tag{4}$$

The validation training curves for the model are shown below, across 5 orders of magnitude of the mixing term λ, and 100 epochs each run. At low mixing, the "inertial" loss is relatively unconstrained, but quickly curves down as the loss mix term increases. Higher values of λ result in higher reconstruction loss, meaning that the output of the decoder is less like the input. However, we will see in the reconstructed video frames that the output remains legible even with a highly constrained "what" encoder.



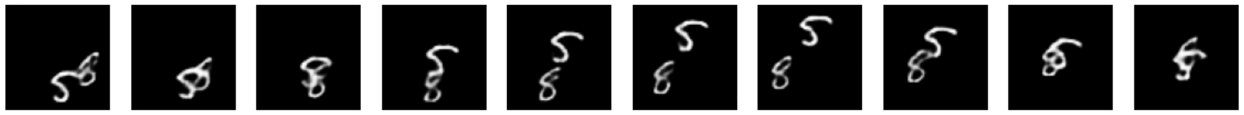
(a) Reconstruction Loss



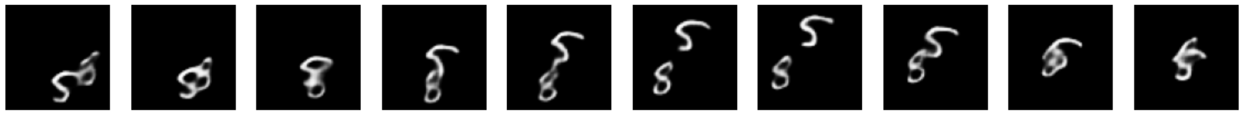
(b) Inertial Loss

Figure 4: Validation Training Curves for Inertial Autoencoder

To check the validity of the autoencoder output, we can display the output of the decoder for 10 different frames of the first video in the validation dataset. We can see that the video (consisting of an '8' and a '5' moving across the 64x64 frame) is faithfully reproduced across the 5 orders of magnitude of λ:



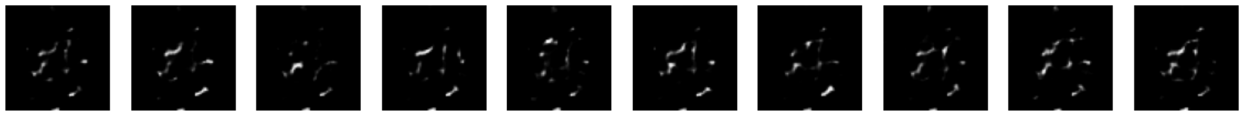
(a) $\lambda = 0.001$



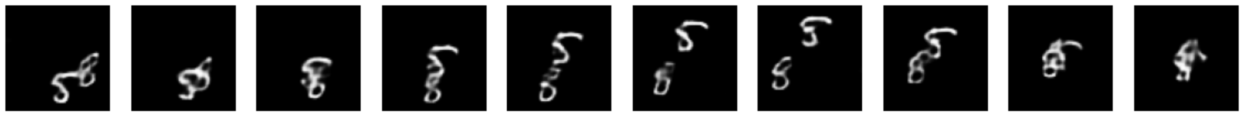
(b) $\lambda = 10.0$

Figure 5: Video Reconstruction for Different Values of Inertial Loss Mixing λ

We can peek "under the hood" of the autoencoder to see the impact of increasing λ, by zero'ing out one or the other encoder. If all of the information required to regenerate the input is carried by either the "what" or the "where" encoder, it should become apparent.



(a) Video reconstruction with only the "where" encoder

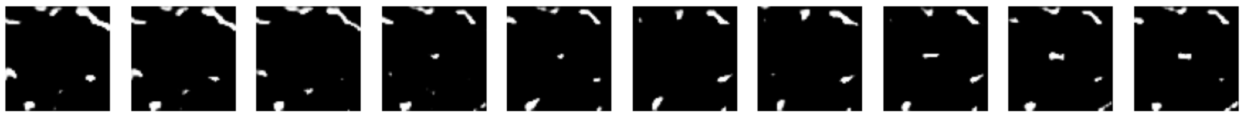


(b) Video reconstruction with only the "what" encoder

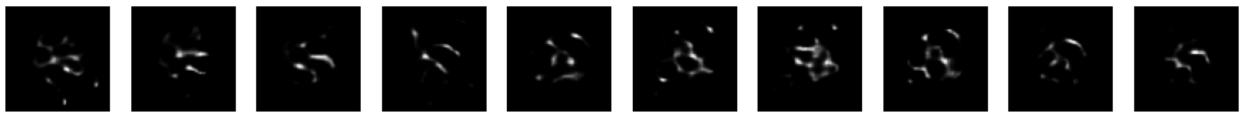
Figure 6: Video Reconstruction with Ablated Encoder for $\lambda = 0.001$

Indeed, we can see that at low levels of λ, the "what" encoder, with its higher-dimensional latent space, carries almost all of the information required to regenerate the input – including the location of the digits in the image. It has failed to disentangle identity from motion at this level of loss mixing.

However, as the loss mixing λ is increased, we can see that the decoder requires the output of both encoders to faithfully reproduce the input. The makes us hopeful that the model has learned to disentangle.



(a) Video reconstruction with only the "where" encoder

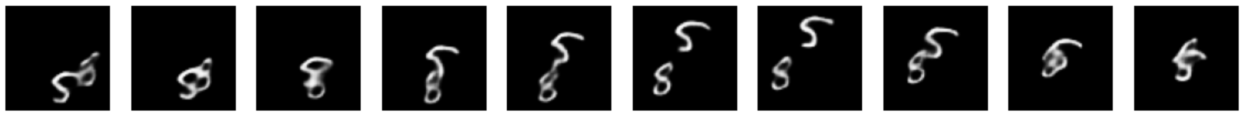


(b) Video reconstruction with only the "what" encoder

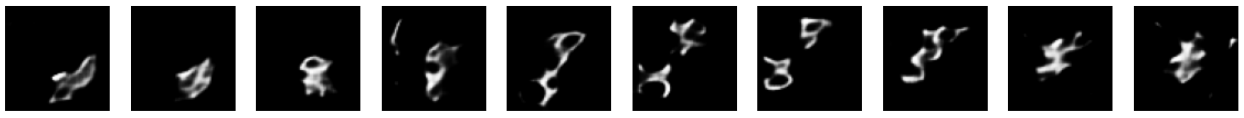
Figure 7: Video Reconstruction with Ablated Encoder for $\lambda = 10.0$

The proof of the pudding, of course, will be if we can swap the embedded vectors of the "what" and the "where" outputs, prior to them being fed into the decoder. If the decoder can generate meaningful results with the scene identity from the "what" encoder of one video combined with the scene motion from the "where" encoder of a different video, then we have shown that we can disentangle motion from identity.

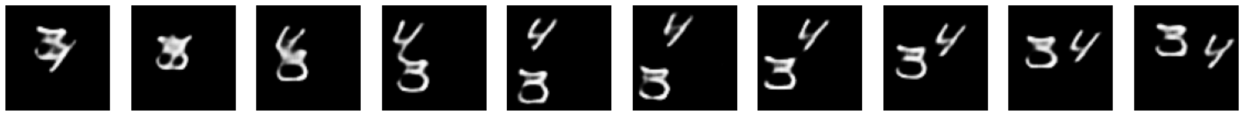
This is shown below, mixing the encoder outputs from two videos: video 1 with a "5" and an "8", and video 2 with a "3" and a "4".



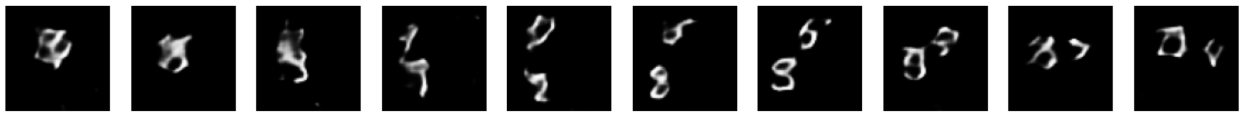
(a) Reconstructed Video 1



(b) "What" Encoder 2 (3 and 4) with "Where" Encoder 1



(c) Reconstructed Video 2



(d) "What" Encoder 1 (5 and 8) with "Where" Encoder 2

Figure 8: Video Reconstruction with Swapped Encoders for $\lambda = 10.0$

The "swapped encoder" configuration appears to faithfully reproduce the motion as intended. The location of the digits in part "a" and "b" are apparently aligned, as are the location of the digits in part "c" and "d" – showing that swapping the "where" encoder produces the intended results.

However, identity of the digits in part "a" and "d" are hard to discern (though they should be the same between the two videos), as are the identity of the digits in part "b" and "c". It is only in the middle of the generated videos that one can start to see a swapped identity. This appears to be due to the fact that the position of the digits in both original videos coincidentally aligned for those frames.

It is worth noting that while the inertial loss term appears to encourage invariance of the "what" encoder's output for consecutive frames of the same video, it does nothing to prevent the encoder's output from being invariant across *all* of the videos. We still need the "what" encoder to produce unique output for different videos. This is where the "Barlow Twins" loss function comes in. [Zbontar et al. [2021]]

3.2 Barlow Twins Loss Function

The Barlow Twins loss term gets its name from neuroscientist H. Barlow's *redundancy-reduction principle* [Barlow et al. [1961]]. The loss term was originally applied to a self-supervised learning problem on static images, in which two identical networks are fed with distorted versions of the same image. The network aims to generate embedding vectors of the distorted images to be as similar as possible, while minimizing the redundancy between different images.

In the context of this paper, the loss term aims to cause the "what" embedding vector to be invariant between adjacent video frames, while reducing the redundancy between the encodings of the frames of other videos.

Barlow Twins is formulated with the elements of the cross-correlation matrix C computed between the outputs z_1 and z_2 for a batch of inputs:

$$L_{\text{BT}} = \underbrace{\sum_i (1 - C_{ii})^2}_{\text{invariance term}} + \underbrace{\alpha \sum_i \sum_{j \neq i} C_{ij}^2}_{\text{redundancy reduction term}} \tag{5}$$

where

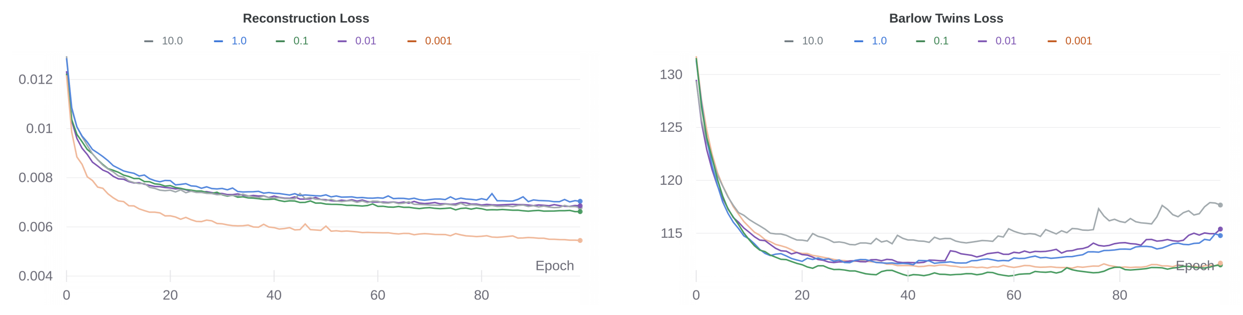
$$C_{ij} = \frac{\sum_b z_{b,i}^A z_{b,j}^B}{\sqrt{\sum_b (z_{b,i}^A)^2} \sqrt{\sum_b (z_{b,j}^B)^2}} \tag{6}$$

C forms a square matrix with size d, where d is the dimensionality of the encoder output ($z \in \mathbb{R}^d$).

The Barlow Twins loss function is combined with the original autoencoder reconstruction loss term, again with a constant mixing term λ, to produce the total loss term use to train the network:

$$L_{\text{Total}} = L_{\text{R}} + \lambda L_{\text{BT}} \tag{7}$$

As before, the network is trained for 100 epochs each run, across 5 orders of magnitude of λ. The Barlow Twins on-diagonal/off-diagonal mixing term α is set to 0.05. As in the previous section, higher values of λ resulted in higher reconstruction loss; though, again, high values of λ still resulted in legible decoder output.

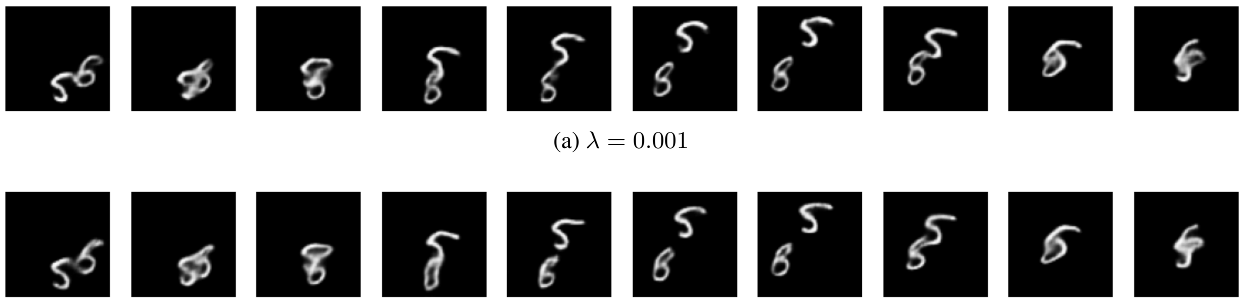


(a) Reconstruction Loss (b) Barlow Twins Loss

Figure 9: Validation Training Curves for Barlow Twins Autoencoder

The network does show signs of overfitting at high λ, with the validation curve rising in the Barlow Twins loss term as training continues past around 30 epochs. Future work could perhaps increase the size of the dataset or explore architectures to minimize overfitting.

Reconstruction fidelity is shown in figure below, showing good reconstruction of the input across many orders of magnitude of λ:

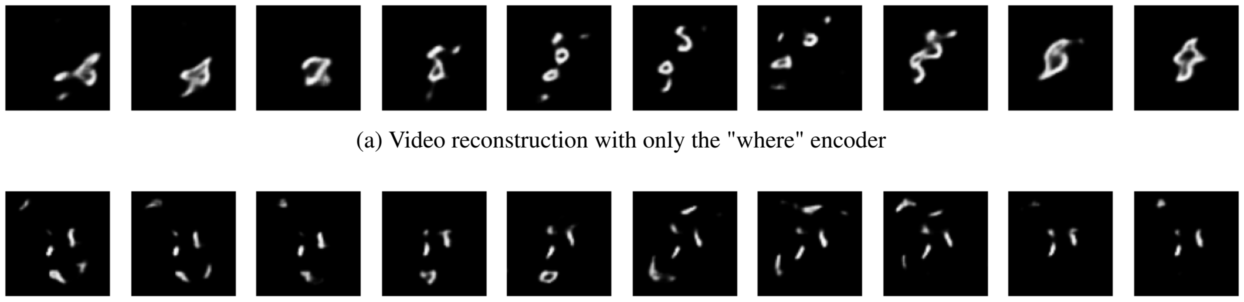


(a) $\lambda = 0.001$

(b) $\lambda = 100.0$

Figure 10: Video Reconstruction for Different Values of Barlow Twins Loss Mixing λ

The encoder ablation study for the Barlow Twins experiment shows much more stable structure for the "what" encoder than was seen with the Inertial Loss term, suggesting that Barlow Twins is doing a better job at deriving a stable "what" vector to describe the video. However, the "where" encoder in the ablation study showed clear signs of encoding specific features of the moving digits, with a "5" and an "8" clearly visible in one of the frames. This means that, despite the low dimensionality of its latent space, the "where" encoder is carrying more of the video identity than was seen in the previous section. Further work would be required to mask the identity of the digits from the "where" encoder.



(a) Video reconstruction with only the "where" encoder

(b) Video reconstruction with only the "what" encoder

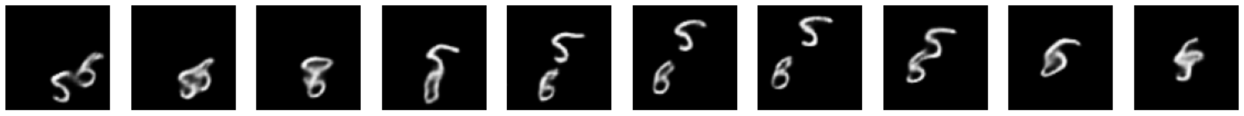
Figure 11: Video Reconstruction with Ablated Encoder for Barlow Twins $\lambda = 10.0$

Finally, we get to the test of swapping the encoders - this time with the Barlow Twins loss function. The results (shown in Figure 12) are as tantalizing and frustrating as in the Inertial loss case. Again, the motion encoding appears to transfer over correctly (this is easier seen in video). The "what" encoder, though, remains stubbornly resistent to inject the identity of one video into the motion of another. It seems likely that the information smuggling discussed in the encoder ablation section is to blame. The decoder is relying too much on the output of the "where" encoder when it reconstructs the image.

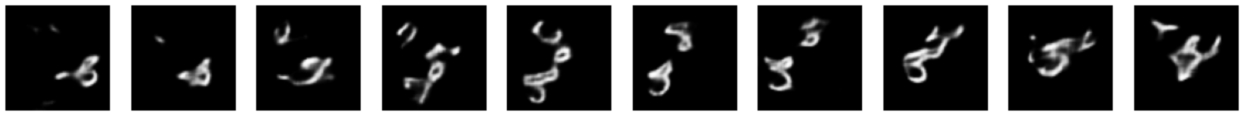
4 Future Direction

The experiments demonstrate partial success in disentangling mtion from identity in the simple videos provided by the MovingMNIST dataset. They also highlight several additional areas to explore.

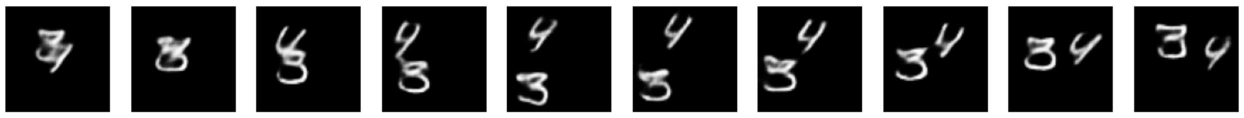
First, techniques could be developed to prevent identity information from being smuggled into the "where" encoder, such as blurring the content or downsizing the resolution of its input. It is worth noting that, in the biological system, the "where" visual stream receives input from a wide field of view with lower resolution, whereas the "what" visual stream receives input from a narrow field of view. Different input to the two systems could enable better disentangling.



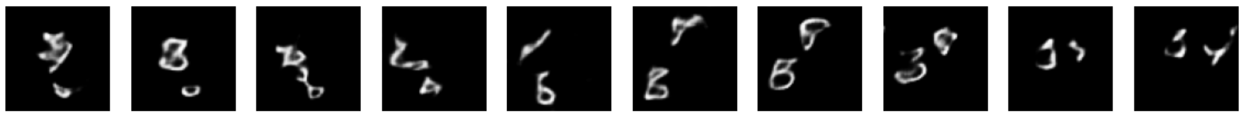
(a) Reconstructed Video 1



(b) "What" Encoder 2 (3 and 4) with "Where" Encoder 1



(c) Reconstructed Video 2



(d) "What" Encoder 1 (5 and 8) with "Where" Encoder 2

Figure 12: Video Reconstruction with Swapped Encoders for Barlow Twins $\lambda = 10.0$

Second, more could be done to ensure the latent space of the encoders produced meaningful results when sampled at points outside of what is generated in the training set. For example, the encoder output could be stochastically sampled by the decoder during training, as is done in the case of variational autoencoders [Kingma and Welling [2014]].

Finally, metrics could be developed to evaluate the success of disentangled representations. While the autoencoder is self-supervised in reproducing the original input, the experiments on swapped encoders were evaluated subjectively. In this regard, metrics similar to what are used to train GANs, such as Frechet Inception Distance [Heusel et al. [2017]] (but for MNIST) could be explored.

References

Lilian Weng. From autoencoder to beta-vae. *lilianweng.github.io/lil-log*, 2018. URL `http://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html`.

Sergiu Oprea, Pablo Martinez-Gonzalez, Alberto Garcia-Garcia, John Alejandro Castro Vargas, Sergio Orts, José Rodríguez, and Antonis Argyros. A review on deep learning techniques for video prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP:1–1, 12 2020. doi:10.1109/TPAMI.2020.3045007.

Melvyn A. Goodale and A.David Milner. Separate visual pathways for perception and action. *Trends in Neurosciences*, 15(1):20–25, 1992. ISSN 0166-2236. doi:https://doi.org/10.1016/0166-2236(92)90344-8. URL `https://www.sciencedirect.com/science/article/pii/0166223692903448`.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852. PMLR, 2015.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

Horace B Barlow et al. Possible principles underlying the transformation of sensory messages. *Sensory communication*, 1(01), 1961.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf`.