The notebook we created for this lab is available in the following GitHub repository: https://github.com/javierferna/multi_view_vae

METHODOLOGY

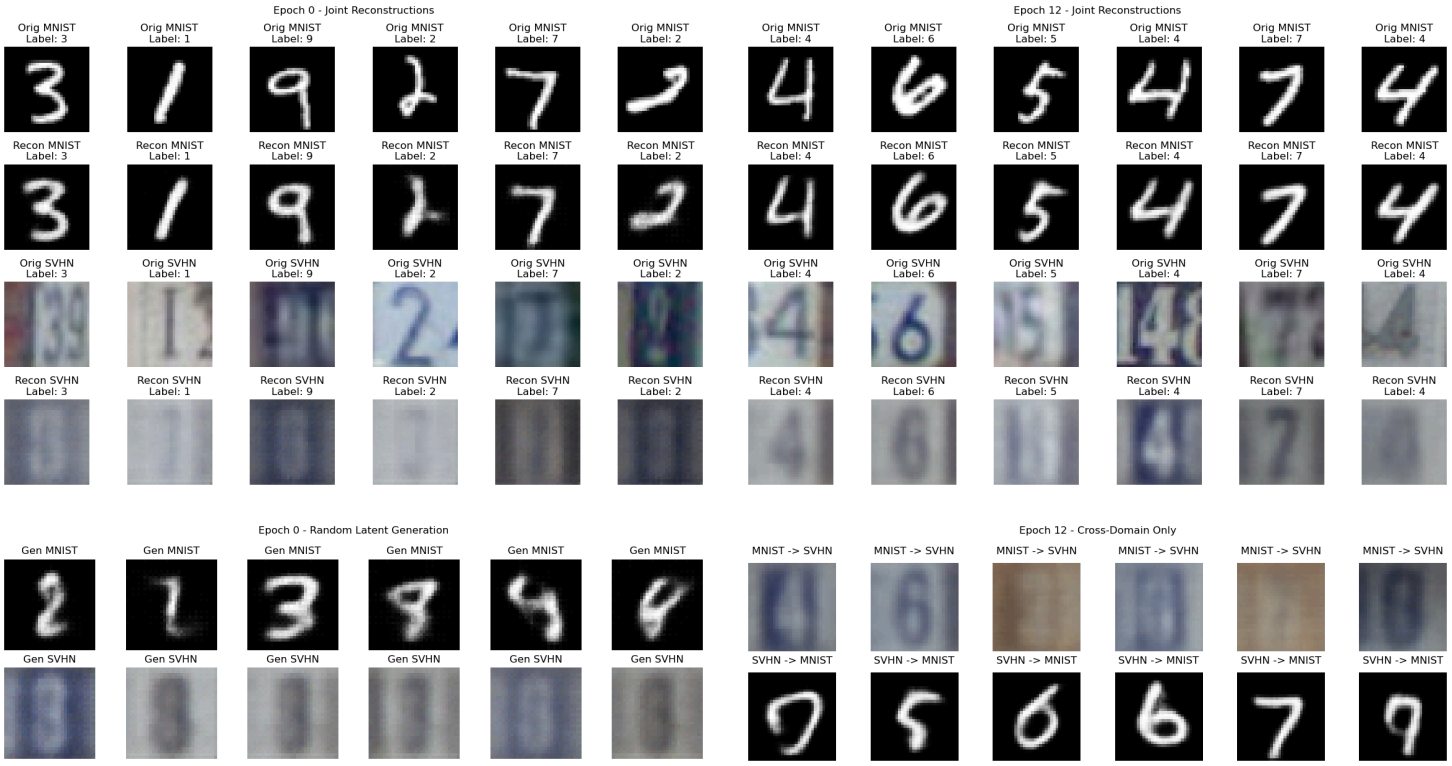
The datasets we are using are readily available using the `torchvision` package from `PyTorch`. In particular, we are working with the **MNIST** and **SVHN** datasets, both of which contain images with numbers. After ensuring both datasets are correctly downloaded, we created a `PairedDatasets` class that allows us to create a database with paired samples from both data sources that have the same label. The final dataset has 56608 paired samples (or 113216 observations).

We implemented a **Multi-View VAE** architecture, which consists of separate CNN-based encoders and decoders for both the **MNIST** and **SVHN** views. These models are designed to map their respective inputs to a shared, common latent space with a dimension of 20.

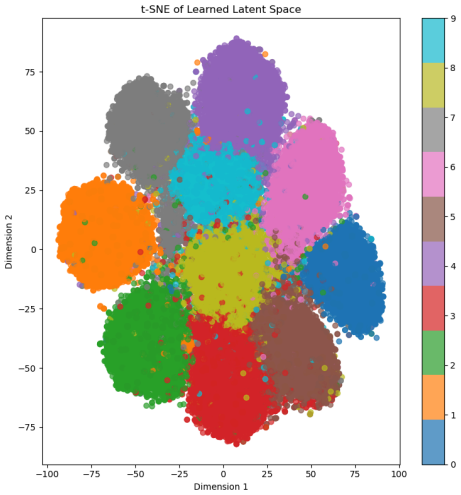
The projection into this shared latent space is made through a **Product of Experts (PoE)**. This method combines the parameters (mean and log-variance) from each encoder to compute a joint posterior distribution. The model was then trained by optimizing the **Evidence Lower Bound (ELBO)**. This loss function is a combination of Binary **Cross-Entropy (BCE)** loss (MNIST), **Mean Squared Error** loss (SVHN), and the **Kullback-Leibler (KL) divergence**, which acts as a regularizer for the latent space.

VISUALIZATIONS

The first set of plots shows the model's performance on **joint reconstruction**, **random latent generation,** and **cross-domain generation** after 1 and 12 epochs of training:



The final plot shows the 20-dimensional shared latent space, projected into 2D using the **t-SNE** method. The points are colored according to their true digit label (0-9). Except for some points in label 3 that scatter, the clusters appear effectively separated.



This indicates that the model has learned a meaningful shared latent representation to distinguish between digits.

RESULTS DISCUSSION

The generations show that the model is successfully learning a shared representation. The **joint reconstructions** are recognizable for both modalities, which confirms the model can encode and decode information from the joint latent space. What's more, the model generates plausible digits in both styles from a single random vector, often representing the same digit.

The **cross-domain generation** also performs really well. The model can take an **SVHN** digit (e.g., a "2"), encode it into the abstract latent space, and then use the **MNIST** decoder to generate a convincing **MNIST** "2". The model has learned a generalized concept of each digit that is independent of the specific view.

The **t-SNE plot** displays this fusion of information. The latent space shows clear, distinct clusters corresponding to the different digit classes. These clusters represent the combined knowledge from both the **MNIST** and **SVHN** datasets, as they are derived from the Product of Experts (PoE) model.

CHALLENGES

At first, we had trouble correctly building the paired dataset. Since the datasets are not paired by default, we had to write a custom `PairedDatasets` class to iterate through both **MNIST** and **SVHN** and match samples based on their labels.

Additionally, we had to implement the main logic of the joint VAE from scratch. This involved creating separate encoder and decoder networks for each modality while ensuring they connected to the shared latent space.

The most critical component was correctly implementing the `product_of_experts` and `vae_loss` functions. This required understanding the mathematical formulation for combining two Gaussian distributions to calculate the parameters of the joint posterior distribution, which is then sampled to produce the shared latent variable z. The loss function includes arguments to consider individual or joint losses for training.

CONCLUSION

This lab has allowed us to learn more about **multi-view VAEs** and their applications to multimodal data. We learned how a **Product of Experts (PoE)** can be used to combine information from two different views into a shared latent space. The results from reconstruction and random generation show that the model learned an abstract representation of the digits, enabling it to translate between the two different data styles.