

TDA Project

Emma Andersdotter - emma.andersdotter@umu.se
Viktor Vigren Näslund - viktor.vigren.naslund@umu.se
Dominik Frey - dominik.frey@liu.se
Anastasja (Nathaniel) Helgesen - nathaniel.helgesen@liu.se
Juliana Francis - jfrancis@kth.se

January 5, 2026

1 Aims and Objectives

In this project we applied the skills and tools we gained from the topological data analysis WASP course to be able to analyse how well variational autoencoders maintain topology within their latent spaces. We work with the MNIST, CIFAR-10 and Concentric Triangles as example datasets to demonstrate this. We describe the theoretical background from the course relevant to this, as well as a background on various VAEs, GMM-VAEs, as well as variants VAEs that have attempted to enforce topological structure directly within their latent space. Within this project we aimed to explore:

- How do the embeddings produced by statistical encoder models produce persistence diagrams and barcodes?
- How effectively can these embeddings be clustered?
- How similar are the persistence diagrams/barcodes to those that can be generated the original data? Is the topology of the space preserved?

Further, we hypothesise that while GMMVAEs (gaussian mixture model variational autoencoders) may be able to maintain clusters within the input data space, H_1 will not be maintained within the latent space.

2 Theoretical Background

2.1 Homology and Persistent Topology in Point Cloud Data

Throughout, we'll work with data of the form

$$X = \{x_1, x_2, \dots, x_n\} \subset \mathbb{R}^d$$

equipped with the distance function $d : X \times X \rightarrow [0, \infty]$. A common way to associate a simplicial complex to (X, d) is through the Vietoris–Rips construction. Given a real number $t \geq 0$, the Vietoris–Rips complex is given by

$$K := \text{VR}_t(X, d) = \{\sigma \subset X \mid d(x, y) \leq t, \forall x, y \in \sigma\}. \quad (1)$$

Fix an order of the vertices and let $C_k(K)$ denote a vector space generated by k -chains in K . We can then define the boundary function $\delta_n : C_n(K) \rightarrow C_{n-1}(K)$ by

$$\delta_n : (v_0 < v_1 < \dots < v_n) \mapsto \sum_{i=0}^n (-1)^i (v_0 < \dots < \hat{v}_i < \dots < v_n),$$

where $\hat{v}_i$ means that v_i is omitted.

The elements in $\ker(\delta_k)$ are called k -cycles and the elements in $\text{im}(\delta_k)$ are called k -boundaries [2]. Since $\delta_k \circ \delta_{k+1} = 0$ [2], every $k+1$ -boundary is also a k -cycle. The k -th homology is defined as

$$H_k(K) = \ker(\delta_k) / \text{im}(\delta_{k+1})$$

and captures equivalence classes of cycles modulo boundaries, i.e., the k -dimensional "holes" of K .

The k -th Betti number is defined as

$$\beta_k = \dim(H_k(K))$$

and gives the number of independent k -dimensional holes of K .

So far, we have considered Vietoris-Rips complexes VR_t where the positive real number t is constant. Increasing t may add additional simplices while previous ones persist, resulting in a monotone sequence of inclusions. We can thus define a *filtration*

$$\text{VR}_{t_1} \subset \text{VR}_{t_2} \subset \dots \subset \text{VR}_{t_m}, \quad t_1 < t_2 < \dots < t_m.$$

The inclusions induce linear maps $H_k(\text{VR}_{t_i}) \rightarrow H_k(\text{VR}_{t_j})$, $i \leq j$. Homology classes might be lost or gained under such mappings. A k -dimensional homology class is said to be *born* at time t_b if it appears in $H_k(\text{VR}_{t_b})$ and *dies* at time t_d if it becomes trivial or merges with an older class at $H_k(\text{VR}_{t_d})$. The collection of birth-death pairs (t_b, t_d) over all k defines the *persistence diagram* in dimension k . [4, Chapter VII]

In section 3.3.4, we compute persistence diagrams by constructing Vietoris-Rips filtrations $\{\mathcal{R}_\epsilon\}_{\epsilon \geq 0}$, which correspond to $\text{VR}_t(X, d)$ with $\epsilon = t$ and $X = \{\mathbf{z}_i\}_{i=1}^N$.

2.2 GMM-VAEs

Our work focused on variational autoencoders (VAEs) and variants of these. Autoencoders consists of two parts, an encoder and a decoder. The decoder compresses high-dimensional data (for example, images, audio, video) into lower dimensions, referred to as embeddings which are located within the latent space of the model. The decoder is then able to reconstruct the original data from this embedding. The encoder and decoder are trained in parallel such that the latent space becomes a representation of the data used within the models training. Standard autoencoders are deterministic, and maps each data point to a distinct point in the latent space. Variational autoencoders instead operate by drawing from a distribution learned by the model, where the model learns the mean and variance for each dimension within the embedding (for example, an embedding, often denoted by $\mathbf{z}$, may be a n -dimensional vector, where there are n means and variances for each dimension) [6]. This prior distribution is enforced on the posterior distribution during training via the Kullback-Leibler divergence loss term, which penalizes the posterior distribution for diverging from the prior. Due to the latent space of the VAE being a continuous distribution, it is possible to sample from the latent space to generate new data points not seen within the original data, allowing for VAEs to act as generative models [9]. Further, by sampling points "nearby" to existing data points, data similar to these points may be generated, though these outputs may be of lower quality [1].

Despite these capabilities however, standard VAEs do not inherently cluster data [5], as the latent space is learned to be a smooth and continuous representation, or manifold, of the training data, which is enforced to be similar to a unimodal prior distribution [6]. Further, there is no explicit mechanism implemented to enforce the VAE to cluster data, and so despite the capabilities mentioned above, there may be different "regions" that produce similar data-points, rather than all being clustered within an area. Due to this lack of clustering, and the enforcement of the gaussian prior to enable compression of data into a smooth and continuous latent space, the VAE may not preserve the topology of the data [7]. This may be due to the compression from high dimensional to low dimensional not being possible depending on the original data's topology, which may cause the model to have to break parts of this to provide a good representation. Also if the dimensions of the latent space is too small, separate features may be combined, or multiple clusters may be represented by the same area of the data. This may also lead to the removal of holes within the data also, as a necessity for the model to flatten the data into the lower-dimensional space [8].

A method proposed to allow for VAEs to have clustering capabilities was that of the Gaussian Mixture Model VAE (GMMVAE) [3]. Instead of sampling from a single distribution per point within the embedding, these models use a mixture of gaussians as the prior instead. This allows the model to learn in an unsupervised manner the distinct clusters representing the data. For example, in a space with k known clusters within the training data, the model can be initialized to use a mixture of k gaussians, which it can then learn to represent the distinct clusters within the training data. Each gaussian has a weight within this framework also, which is used to affect the influence of the gaussians outputs on the final embedding. To further enforce clustering, we utilize expectation-maximization optimization during

training, which computes how likely a point should belong to a given gaussian within the mixture, and then adjusts each of the gaussians parameters (mean, variance, and weight) based on these likelihoods. This is continuously updated at each training step. It is possible that the GMMVAE may help to resolve the issues of broken topology brought about via the lack of clustering within the standard VAE, and within this project we aimed to assess this.

2.3 Higher-Order Homology and GMM-VAEs

A GMM-VAE may encourage separation between latent modes, which roughly reflects the number of connected components (H_0) in the input data. However, each latent component is encouraged to be a Gaussian "blob", so higher-order features such as loops are expected to be collapsed. We therefore hypothesize that, while GMM-VAEs are useful for clustering, they may destroy higher-order homological features H_k , where $k \geq 1$.

More explicitly, let X denote the original point cloud, Z the latent representation and β_k the k -th Betti number. We hypothesize that

$$\beta_0(X) \approx \beta_0(Z)$$

and

$$\beta_k(X) > \beta_k(Z)$$

for $k \geq 1$.

2.4 Distances on bar codes

Given a filtration of simplicial complexes A , define $D_k(A)$ as the persistence diagram of homology group k for A . With this we can define bar code distances between $X_k := D_k(X)$ and $Y_k := D_k(Y)$ of two filtrations X and Y . The q-Wasserstein distance is defined as

$$d_W^q(X_k, Y_k) = \inf_{\eta: X_k \rightarrow Y_k} \sum_{x \in X_k} \|x - \eta(x)\|_\infty^q \quad (2)$$

and the bottleneck distance

$$d_\infty(X_k, Y_k) = \inf_{\eta: X_k \rightarrow Y_k} \sup_{x \in X_k} \|x - \eta(x)\|_\infty. \quad (3)$$

For any k these distances themselves define distances between the full persistence diagrams on of X and Y . But we could of course also combine the distances for each homology group. For instance

$$d_1(X, Y) = d_W^q(X_0, Y_0) + d_W^q(X_1, Y_1) + d_W^q(X_2, Y_2) \quad (4)$$

combining the first 3 homology groups.

2.5 InvMap-VAEs

InvMap-VAEs build open standard VAEs by adding an additional loss term during model training to enforce the latent representation to be geometrically and topologically similar to a provided embedding. The provided embeddings used as references are found using dimensionality reduction techniques (e.g. Isomap, t-SNE, or UMAP). The decoder of the network finds a probabilistic inverse mapping from these embeddings back to the original datapoints which is not usually possible using certain dimension reduction methods. To do this, during training the standard VAE loss terms are augmented with an additional loss, which is the mean square error between the VAEs embedding and the provided embedding [7]

2.6 Witness Simplicial VAEs

Witness Simplicial VAEs [7] do not use provided embeddings, but instead attempt to build a witness complex from the original data, and attempt to preserve this structure within the VAEs latent space. Witness complexes are representations of the data that aim to capture the underlying topology while only using a subset of datapoints, known as landmarks. In the case of Witness Simplicial VAEs, these landmarks were chosen arbitrarily, though other methods may be used. Following this, to build the witness complex a radius must be defined (to define a ball surrounding each point within the witness complex). A suitable radius can be found through ensuring the Betti numbers of the overall dataset

match that of the witness complex. A further consideration for these is to ensure the dimension of the simplices are lower or equal to the dimensions of the latent space of the VAE, such that the VAE does not need to learn how to compress the data, and thus potentially break the topology. Following this, two additional terms are added to the loss when training Witness Simplicial VAEs, which aim to enforce the encoder and decoder of the VAE to become simplicial maps. This is done by:

$$L_{SE}(e, K, \alpha) = \sum_{\sigma \in K} \mathbb{E}_{\lambda_j \sim \text{Dir}(\dim(\sigma)+1, \alpha)} \text{MSE} \left(e \left(\sum_{j=0}^{\dim(\sigma)} \lambda_j \sigma_j \right), \sum_{j=0}^{\dim(\sigma)} \lambda_j e(\sigma_j) \right)$$

$$L_{SD}(d, K, \alpha) = \sum_{\sigma \in K} \mathbb{E}_{\lambda_j \sim \text{Dir}(\dim(\sigma)+1, \alpha)} \text{MSE} \left(d \left(\sum_{j=0}^{\dim(\sigma)} \lambda_j e(\sigma_j) \right), \sum_{j=0}^{\dim(\sigma)} \lambda_j d(e(\sigma_j)) \right)$$

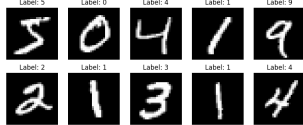
Where LSE and LSD are loss for the encoder and decoder respectively, e and d represent the encoder and decoder, $\mathbb{E}_{\lambda_j \sim \text{Dir}(\dim(\sigma)+1, \alpha)}$ represents the expectation for $(\lambda_j)_{j=0, \dots, \dim(\sigma)}$ following a symmetric dirchlet distribution. α is set to 1 allowing for a uniform distribution over the simplex. By adding these additional terms to the loss function, the model is forced to learn how to construct simplicial maps between the input space to the latent space, and then back from the latent space to the reconstruction space [7].

3 Experiments

3.1 Datasets

The datasets used for evaluation are MNIST, CIFAR-10, and Concentric Triangles.

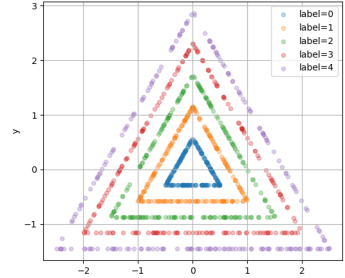
- MNIST: a dataset of 28x28 grayscale images of handwritten digits (see Figure 1a).
- CIFAR-10: a dataset of 32x32 color images of objects from 10 classes (see Figure 1b).
- Concentric Triangles: a dataset of 2x2 images of concentric triangles (see Figure 1c).



(a) MNIST dataset example.



(b) CIFAR-10 dataset example.



(c) Concentric Triangles dataset example.

Figure 1: Dataset examples.

3.2 Baselines

- **RAW:** Raw data.
- **UMAP:** Uniform Manifold Approximation and Projection of the raw data with a 2D latent space.
- **PCA-UMAP:** Principal Component Analysis of the raw data with a latent space dimension of the same size as the embedding dimension of the GMM-VAE model. and then UMAP applied on the PCA embeddings with a 2D latent space.

3.3 Evaluation Setup

The evaluation process consists of three main stages: model training, embedding generation, and topological analysis. For each dataset, we train a GMM-VAE model using a convolutional neural network architecture (GMMCNN) for image datasets (MNIST and CIFAR-10), and a multilayer perceptron architecture (GMM-VAE) for the synthetic concentric triangles dataset.

3.3.1 Model Training

All models are trained using the β -VAE objective function, which balances reconstruction quality with latent space structure:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \beta \cdot \text{KL}(q(z|x)||p(z)), \quad (5)$$

where $\mathcal{L}_{\text{recon}}$ is the mean squared error reconstruction loss, $\beta \in [0, 0.5]$ is the KL weight parameter, and $\text{KL}(q(z|x)||p(z))$ is the Kullback-Leibler divergence between the variational posterior $q(z|x)$ and the learnable GMM prior $p(z) = \sum_{k=1}^K \pi_k \mathcal{N}(z|\mu_k, \Sigma_k)$ with K mixture components.

To facilitate stable training and promote meaningful latent representations, we employ a linear β annealing schedule. The KL weight β starts at $\beta_{\text{start}} = 0.0$ for warmup epochs, then linearly increases over $\beta_{\text{annealing}}$ epochs to reach $\beta_{\text{end}} = 0.005$. This strategy allows the model to first learn good feature representations with emphasis on reconstruction, then gradually incorporate the GMM clustering structure into the latent space.

For MNIST, we train for 80 epochs with a batch size of 32, using Adam optimizer with learning rate $\alpha = 10^{-5}$ and no weight decay. The embedding dimension is set to $d = 10$, which provides sufficient capacity for the 10-class digit separation task. β annealing occurs over 50 epochs.

For CIFAR-10, we train for 80 epochs with the same configuration as for MNIST.

For the ConcentricTriangles dataset, we train for 80 epochs with embedding dimension $d = 5$, matching the number of classes ($K = 5$). All other training hyperparameters match the MNIST configuration. The choice of a lower-dimensional embedding reflects the simpler geometric nature of this synthetic dataset.

3.3.2 Embedding Generation

Following training, we generate multiple types of embeddings from each trained model to enable comprehensive topological analysis. We sample a diverse subset of $N = 2000$ data points from each training dataset, ensuring balanced representation across all classes through stratified sampling. This sampling strategy prevents bias toward dominant classes and ensures that topological features represent the full dataset structure.

For each sampled data point $\mathbf{x}_i$, we compute four types of embeddings:

1. **Raw data:** Flattened pixel values $\mathbf{x}_i^{\text{raw}} \in \mathbb{R}^D$, where D is the product of image dimensions and channels (e.g., $D = 784$ for MNIST, $D = 3072$ for CIFAR-10).
2. **GMM-VAE latent embeddings:** The mean of the variational posterior distribution $\boldsymbol{\mu}_i = \text{Encoder}(\mathbf{x}_i)$, where $\boldsymbol{\mu}_i \in \mathbb{R}^d$ and d is the embedding dimension. We use the mean rather than reparameterized samples to obtain deterministic embeddings that reflect the learned representation structure without sampling noise.
3. **PCA embeddings:** Principal component analysis is applied to the raw data, extracting d principal components to match the GMM-VAE embedding dimension. This provides a linear dimensionality reduction baseline: $\mathbf{z}_i^{\text{PCA}} = \mathbf{W}_{\text{PCA}}^T \mathbf{x}_i^{\text{raw}}$, where $\mathbf{W}_{\text{PCA}} \in \mathbb{R}^{D \times d}$ contains the top d eigenvectors of the data covariance matrix.
4. **UMAP projections:** We apply Uniform Manifold Approximation and Projection (UMAP) to reduce high-dimensional embeddings to 2D for visualization. Specifically, we compute UMAP projections of (a) raw data, (b) PCA embeddings, and (c) GMM-VAE embeddings. For GMM-VAE embeddings with dimension $d = 2$ or $d = 3$, we use the first two dimensions directly without UMAP to avoid unnecessary nonlinear distortion. The UMAP embedding is learned with a random state fixed at 42 for reproducibility.

The choice of using the mean $\boldsymbol{\mu}_i$ rather than reparameterized samples $\mathbf{z}_i = \boldsymbol{\mu}_i + \boldsymbol{\epsilon} \odot \sqrt{\boldsymbol{\sigma}_i^2}$ (where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$) for GMM-VAE embeddings ensures that the topological analysis captures the deterministic structure learned by the encoder, rather than introducing stochastic variation that could obscure meaningful topological features.

3.3.3 2D UMAP Visualization and Comparison

To enable visual inspection of the learned representations and do qualitative comparison between different embedding methods, we generate 2D UMAP projections for each embedding type. For each dataset, we create a three-way comparison visualisation by projecting three embedding types into 2D space using UMAP: (1) raw data, (2) PCA embeddings, and (3) GMM-VAE latent embeddings.

3.3.4 Topological Data Analysis

For each embedding type, we compute persistence diagrams using the Vietoris-Rips filtration, which captures the topological structure of the point cloud across multiple scales. Given a set of N points $\{\mathbf{z}_i\}_{i=1}^N$ in $\mathbb{R}^d$, we construct a sequence of simplicial complexes $\{\mathcal{R}_\epsilon\}_{\epsilon \geq 0}$ where $\mathcal{R}_\epsilon$ contains all simplices whose vertices are pairwise within distance ϵ .

Persistence diagrams encode the birth and death times of topological features (connected components, loops, voids) as the filtration parameter ϵ increases. For each homology dimension $k \in \{0, 1, 2\}$, we extract persistence intervals (b, d) where b is the birth time (radius at which a feature appears) and d is the death time (radius at which it disappears or merges). We limit our analysis to maximum homology dimension $k_{\max} = 2$ to capture connected components ($k = 0$), loops ($k = 1$), and voids ($k = 2$), which are the most interpretable topological features for high-dimensional embeddings.

To manage computational complexity for large point clouds, we subsample to a maximum of $N_{\max} = 500$ points when computing persistence diagrams. Subsampling is performed via random selection with fixed random seed for reproducibility.

From the persistence diagrams, we generate persistence diagrams, which plot points (b, d) in the birth-death plane to reveal the scale at which topological features emerge and persist.

3.4 Results

3.4.1 UMAP Visualization Comparison

MNIST The UMAP visualizations in Figure 2 show that all three embedding methods achieve good cluster separation for the MNIST dataset. The raw data projection displays separated clusters with boundaries between digit classes, suggesting that the original pixel space contains structure that supports class distinction. The PCA embedding preserves this separation, indicating that linear dimensionality reduction maintains the geometric relationships present in the raw data. The GMM-VAE embedding appears to produce more compact cluster formations compared to the other methods, with tighter groupings and reduced spread. This suggests that the nonlinear transformation learned by the GMM-VAE may provide some benefit for cluster compactness, though all three methods achieve adequate separation for this dataset.

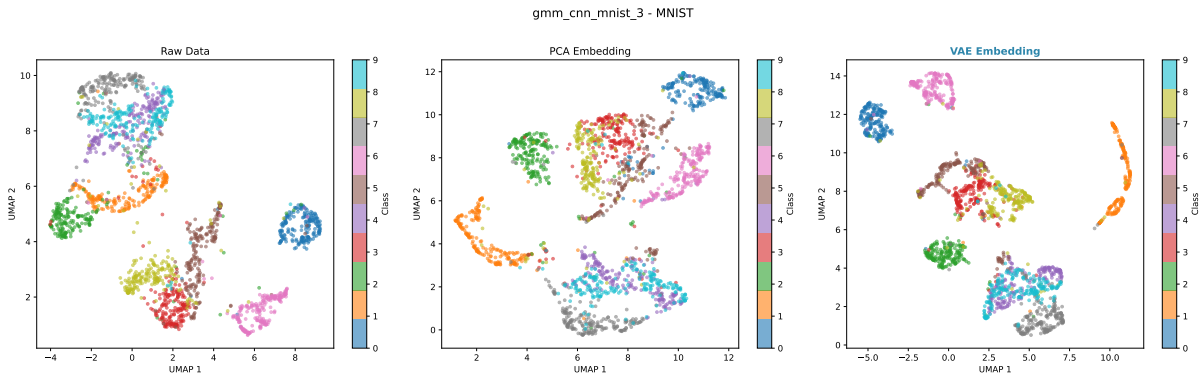


Figure 2: UMAP visualization comparison for MNIST dataset.

CIFAR-10 The CIFAR-10 visualizations in Figure 3 show limited cluster separation across all three embedding methods, with overlap between classes. The raw data projection displays intermingled clusters, which may reflect shared visual features across object classes in natural images. The PCA embedding does not appear to improve separation compared to the raw data, suggesting that linear dimensionality reduction may not capture the relationships needed to distinguish between classes. The GMM-VAE embedding shows similar levels of overlap and boundary ambiguity as the other methods, indicating that

the learned nonlinear transformation does not provide clear improvement in cluster separation for this dataset.

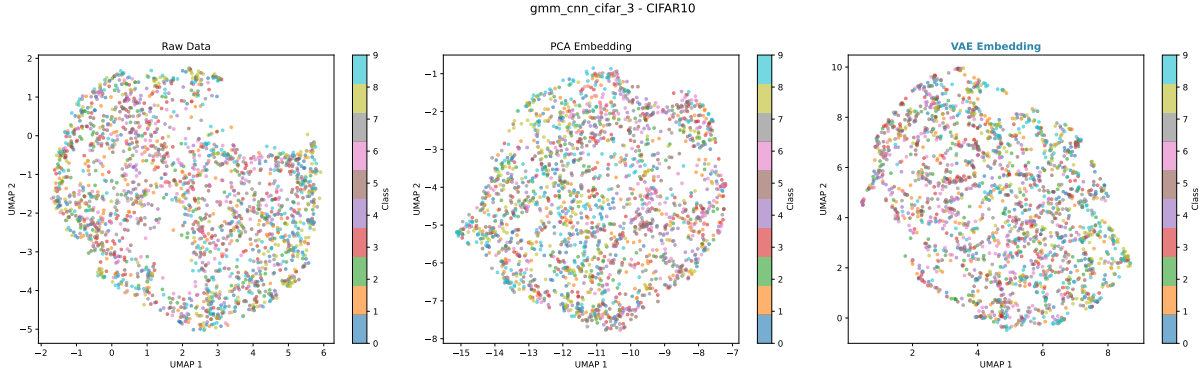


Figure 3: UMAP visualization comparison for CIFAR-10 dataset.

Concentric Triangles The Concentric Triangles dataset in Figure 4 shows good cluster separation across all three embedding methods. The raw data projection displays separated clusters corresponding to the five triangle classes, with boundaries present despite some variation within clusters. The PCA embedding maintains this separation, suggesting that linear transformation preserves the geometric structure of the synthetic dataset. The GMM-VAE embedding appears to produce more compact cluster formations with tighter groupings compared to the other methods. This indicates that the GMM-VAE’s nonlinear transformation may provide some benefit for cluster compactness.

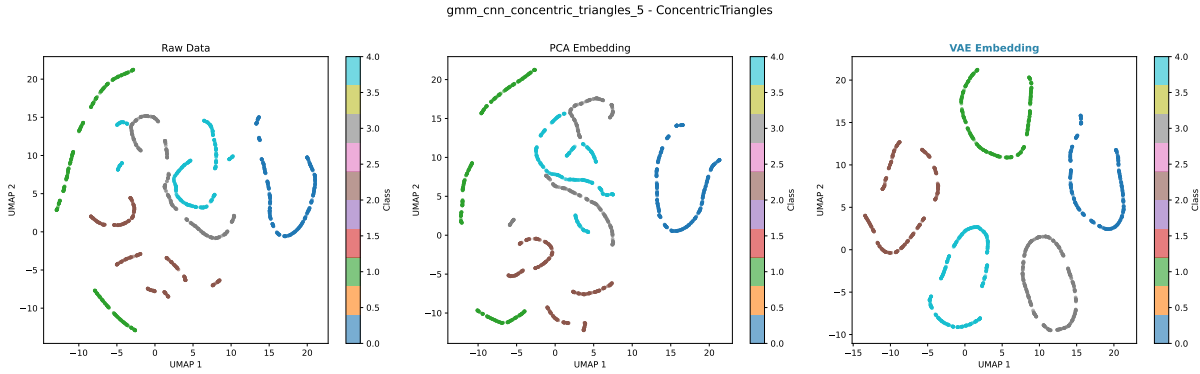


Figure 4: UMAP visualization comparison for Concentric Triangles dataset.

3.4.2 Persistence diagram comparison

In persistence diagrams, distance from the diagonal indicates feature persistence (significance). Points far from the diagonal represent robust topological features. Points near the diagonal are typically noise.

MNIST In Figure 5 we can see the persistence diagrams of various embeddings for the MNIST dataset.

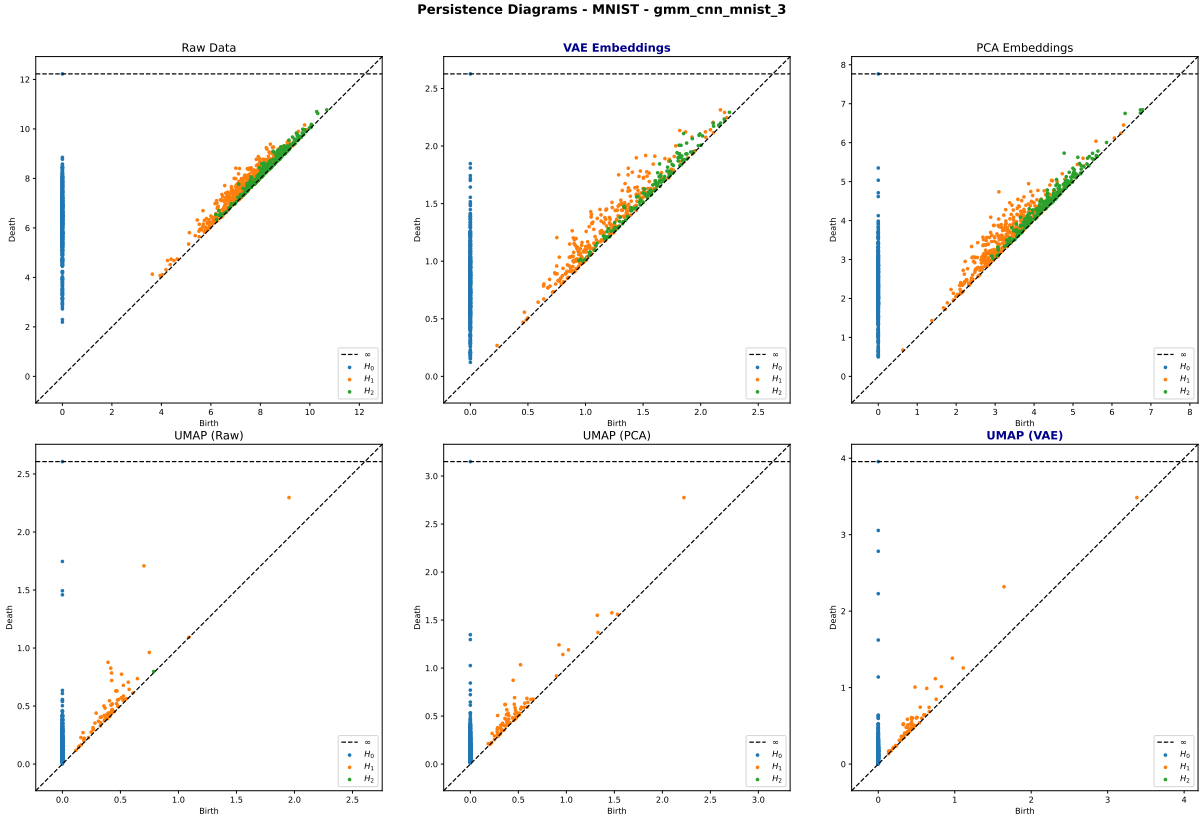


Figure 5: Persistence diagram comparison for MNIST dataset.

Method	Impact on persistence
Raw	Shows a high density of low-persistence noise points along the diagonal.
GMM-VAE	The H_1 (loops) and H_2 (voids) points are more tightly clustered and shifted toward lower birth/death values compared to Raw data. This indicates the GMM-VAE's latent space is smoother and more compact, effectively filtering out some high-dimensional noise.
UMAP (Raw/PCA)	UMAP drastically changes the scale and significantly separates H_0 points. It captures a few highly persistent H_1 features while suppressing the background cloud of low-persistence noise, creating a cleaner topology.
UMAP (GMM-VAE)	This combination produces the most denoised persistence diagram. It shows a clear separation between a few significant topological features and the diagonal. The H_0 points are distinct, implying that UMAP on GMM-VAE embeddings creates the most topologically well-separated clusters of all baselines.

CIFAR-10 and Concentric Triangles In Figure 6 and Figure 7 we can see the persistence diagrams of various embeddings for the CIFAR-10 and Concentric Triangles datasets.

Persistence Diagrams - CIFAR10 - gmm_cnn_cifar_3

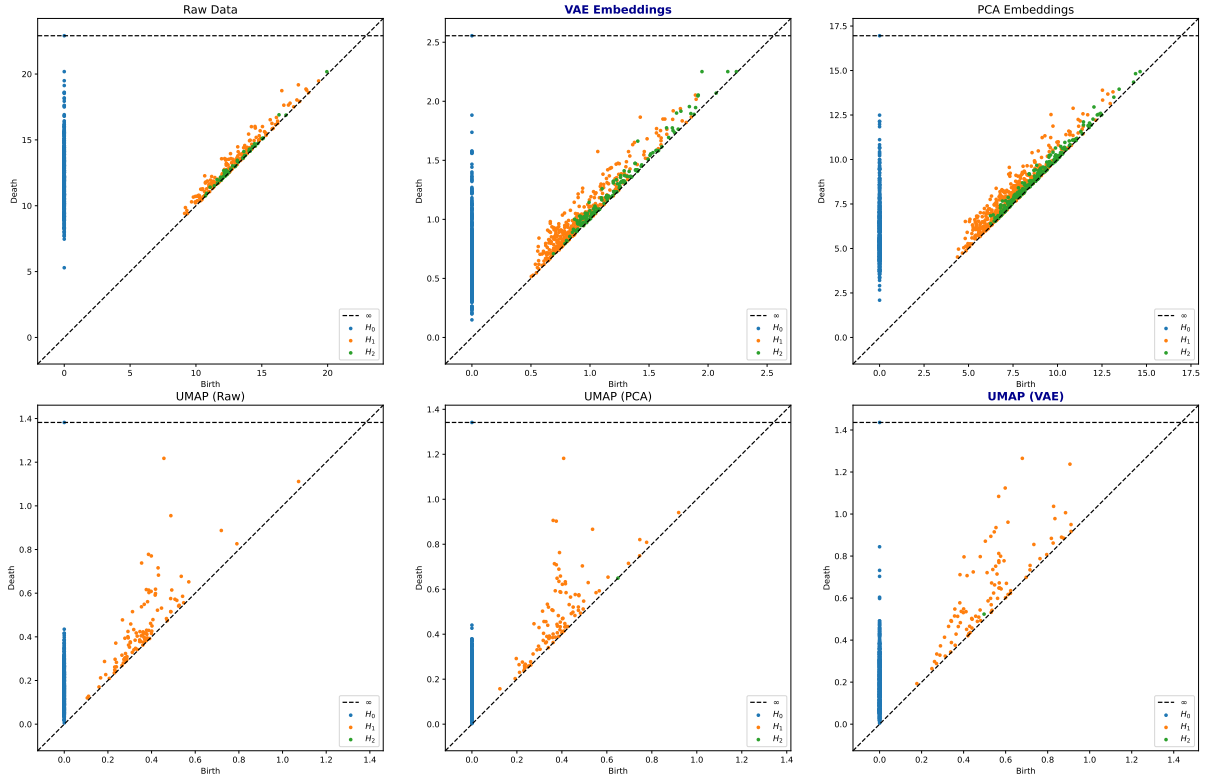


Figure 6: Persistence diagram comparison for CIFAR-10 dataset.

Persistence Diagrams - ConcentricTriangles - gmm_cnn_concentric_triangles_5

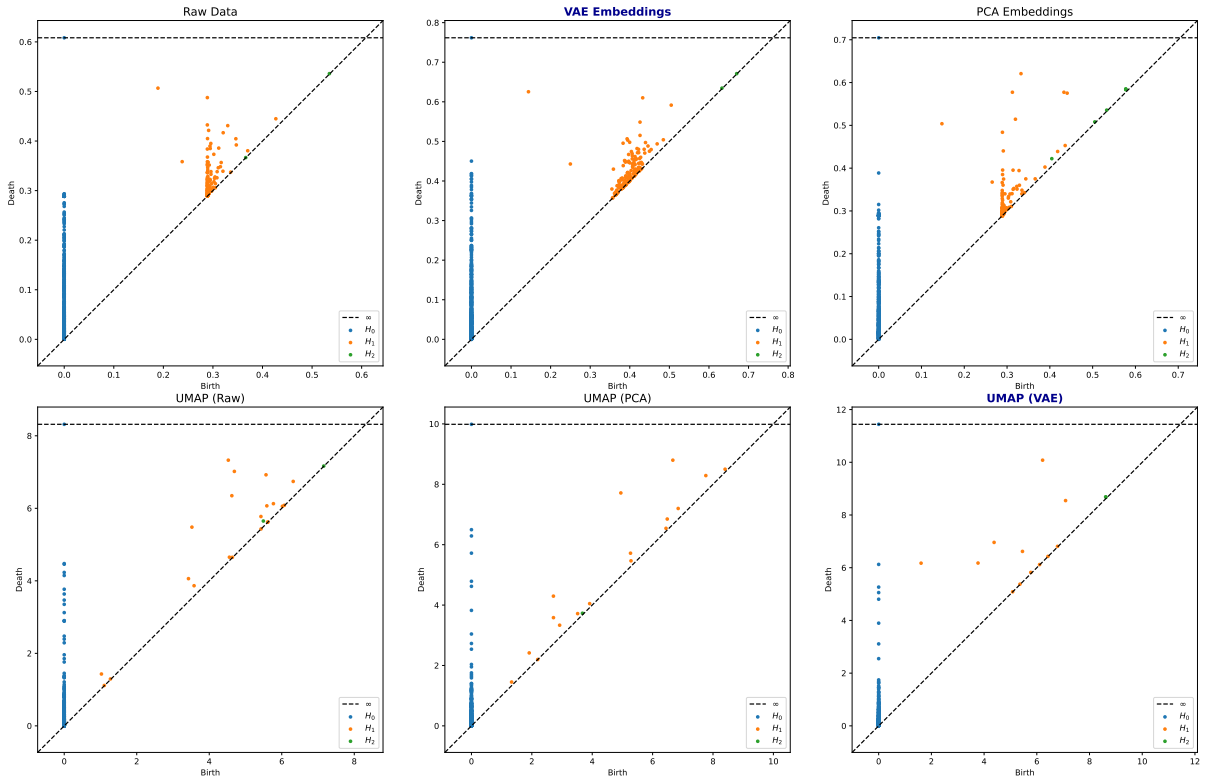


Figure 7: Persistence diagram comparison for Concentric Triangles dataset.

Feature	CIFAR-10	Concentric Triangles
Topology	Dominated by H_0 (clusters) with high H_1 noise.	Strong, persistent H_1 loops.
GMM-VAE effect	Filters noise and reduces global scale.	Preserves and cleans known geometry.
UMAP (GMM-VAE) behavior	Struggles to find persistent loops but good at cluster separation (H_0).	Isolates the ideal topological shape.

3.4.3 Concentric Triangle Analysis in 2-Dimensional Embedding

To show that using the GMM-VAE for latent space regularization disrupts higher order features in datasets that contains such, we take a simplified case using our concentric triangles dataset. We train four models with gradually increasing β value that input 2 dimensional embeddings and plot the inferred values of our data without using UMAP on them.

In Figure 8 we can see the 2D VAE embedding of the concentric triangles dataset with different β values (0.001, 0.01, 0.1, 1.0) along with the reconstruction of the original data from the embeddings.

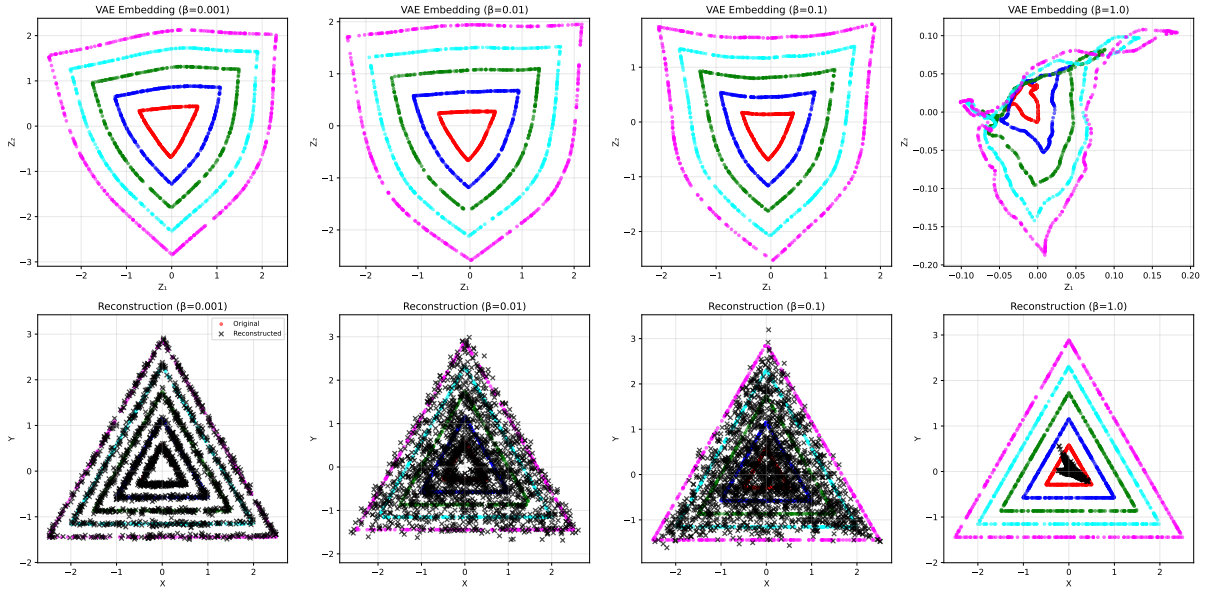


Figure 8: 2D embedding of the concentric triangles dataset.

As the β parameter increases in GMM-VAE, the model shifts from topological reconstruction to latent space regularization. This transition has a visible impact on the loops in the persistence diagrams. Increasing β forces the latent representation to fit a Gaussian prior, which effectively fills in the empty centers of the triangles. While the embedding becomes more separated into clusters, the reconstruction loses the sharp loops present in the original data. At high β (e.g., 1.0), the H_1 points would disappear or have extremely low lifetimes, indicating that the embedding has collapsed into solid blobs rather than loops.

3.4.4 Distance on persistence diagrams

In order to more quantitatively understand if the persistence diagrams the GMM-VAE produces are actually different from, for instance the PCA encoding, we do clustering on the persistence diagrams. We choose 4 VAEs with different choices of β . We choose 22 subsets of MNIST, each consisting of 500 points. Every subset gets one persistence diagram for $H_k, k \in 0, 1, 2$ and for each VAE model and the PCA method. We then compute distances between all pairs of persistence diagrams. We compute 4 distances: the 1-Wasserstein distance for the persistence diagram for each k and then the sum of these, i.e., (2) for $k \in 0, 1, 2$ and then (4).

The distance matrices between the persistence diagrams can be seen in Fig. 9

To see if there are any clear clusters, or clear cluster separation, between PCA and VAE we, produce several cluster plots from our distance matrices. We use MDS, t-SNE and UMAP. See Fig. 10. As can be

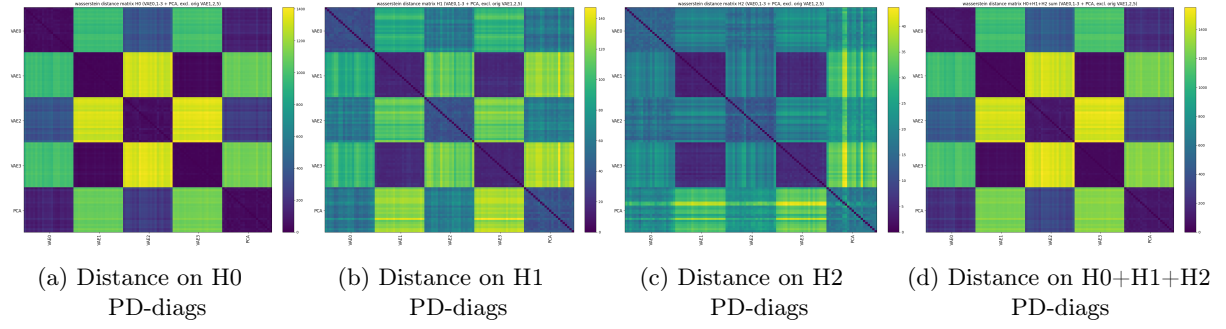


Figure 9: The four distance matrices, Wasserstein on H0, H1 and H2, then finally their sum. Each visible block is where one VAE model compares to another (or itself or PCA). Each block consists of the 22 subdatasets

seen in the figures, there is quite clear separation between the models and PCA using all distances and all representations (MDS, t-SNE and UMAP).

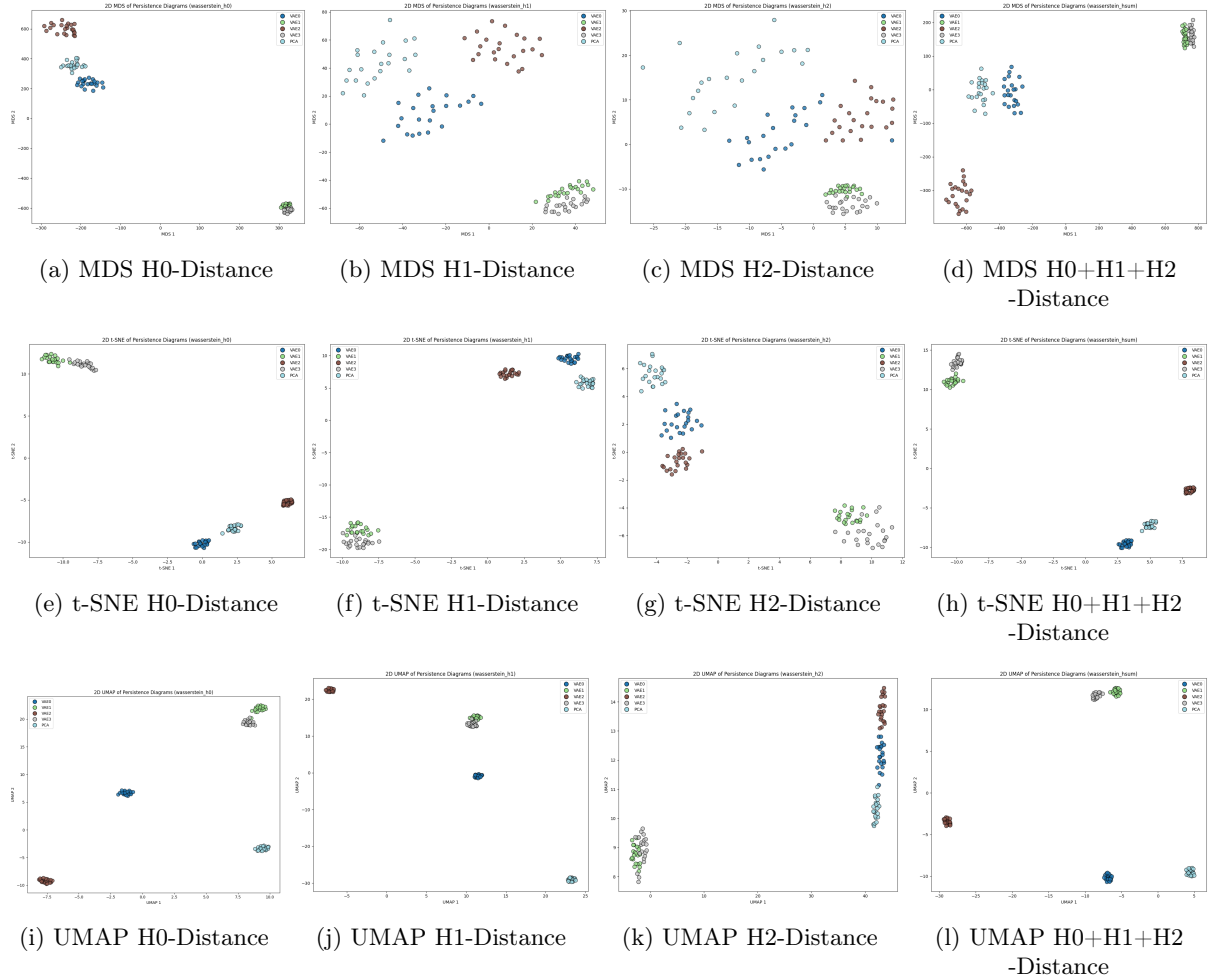


Figure 10: Cluster plots for MDS, t-SNE and UMAP and the 4 distances. Plots show quite clear separation between models in all 4 distances, but no clear distance between PCA and all other GMM-VAE

3.5 Takeaways

The GMM-VAE + UMAP pipeline acts as a topological filter regardless of the dataset. While the RAW data is noisy and lacks clear separation of features, UMAP (GMM-VAE) isolates the core persistent

features, suggesting that the GMM-VAE successfully learns a representation that UMAP can then project with strong topological properties. But this depends on the underlying representation. For synthetic data, it uncovers the true shape, for complex images like CIFAR-10, it simplifies the representation so much that higher-order features (H_1, H_2) often vanish, leaving only a well-separated cluster structure (H_0).

In terms of loop (and possibly higher order feature) persistence, the trade off between regularisation over reconstruction for the synthetic dataset causes the H_1 points to collapse. Consequently, while the embedding achieves better cluster separation, it suffers a total loss of topological persistence for the loops that defined the original geometry.

While it is difficult to visually assess whether the persistence diagrams produced by the PCA and VAE models differ, quantitative analysis based on persistence diagram distances and clustering indicates that each model generates a somewhat distinct set of diagrams. However, no clear separation between the GMM-VAE models as a group and PCA can be established.

4 Conclusion

We found that the VAE was able to isolate and cluster core features within the datasets we assessed our work on, which suggests that VAEs are able to at least represent certain topological features within its latent space. In contrast to this however, our results do imply that higher order features such as H_1 and H_2 are broken within the latent space of the VAE, suggesting that not all features are maintained. Further, our results on VAEs found that birth/deaths occurred at lower values than in the raw data, which implies that the latent space of the VAE is smoother and more compact, which is in line with prior work and our assumptions. When assessed on UMAP embeddings of the input, H_0 points are still distinct, even if other features still appear to have been destroyed.

Through participating in the WASP TDA course, we acquired many valuable skills that may be applicable in the rest of our graduate studies. We also learnt how common machine learning models used may be breaking data representations, and gave us insight that perhaps there could be work to be done to improve or enhance to improve the capabilities of these through methods learnt within this course.

References

- [1] Alejandro Ascarate, Leo Lebrat, Rodrigo Santa Cruz, Clinton Fookes, and Olivier Salvado. Improving the generation of vae’s with high dimensional latent spaces by the use of hyperspherical coordinates. *arXiv preprint arXiv:2507.15900*, 2025.
- [2] Wojtek Chacholski. Topological data analysis, meeting 2 [lecture slides]. Course Handout, 2025. Module 2 in TDA, 2025.
- [3] Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.
- [4] Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2010.
- [5] Tiffany Fan, Murray Cutforth, Marta D’Elia, Alexandre Cortiella, Alireza Doostan, and Eric Darve. Physically interpretable representation learning with gaussian mixture variational autoencoder (gm-vae). *arXiv preprint arXiv:2511.21883*, 2025.
- [6] Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.
- [7] Aniss Aiman Medbouhi. Towards topology-aware variational auto-encoders: From invmap-vae to witness simplicial vae, 2022.
- [8] Luis A Perez Rey, Vlado Menkovski, and Jacobus W Portegies. Can vae’s capture topological properties? In *NeurIPS 2019 Workshop*, 2019.
- [9] Sanchayan Vivekananthan. Comparative analysis of generative models: Enhancing image synthesis with vae’s, gans, and stable diffusion. *arXiv preprint arXiv:2408.08751*, 2024.