AUTOENCODERS & CONTROLLING LATENT SPACE FOR GENERATIVE MODELING

Maria DaRocha

VUW, AIML425: Neural Networks and Deep Learning

1. INTRODUCTION

The following is my own work and supported by code available on GitHub and Google Collab.

The following experiment aimed to develop an autoencoder (non-VAE) capable of reconstructing 3D data uniformly distributed over the surface of a cube. Once trained, the model was tested in a generative setting. The study was then extended to a novel investigation on the effect of competing latent regularisation terms using annealing and Bayesian optimisation. *Figure 1* (*Appendix*) provides a visual reference of the target data. The experiments were set up using Python and Jupyter Lab.

2. THEORY

This paper defers to "*Demonstrating the Basic Principles of Generative Models for (FCNN) Gaussian and Uniform Mappings*" for its discussion on objective functions, regularisation, maximum mean discrepancy (MMD), Gaussian kernels, and sigma opimisation using the median heuristic. [1]

Further theoretical background will outline Kullback-Leibler (KL) divergences, autoencoders, probabilistic formulation of the Mean Square Error (MSE) objective function, variational autoencoders (VAEs), annealing, and Bayesian optimisation.

KL-divergences are asymmetric and non-negative measures of relative entropy that quantify the divergence of one probability distribution (often a model's prediction, q) from a reference or ground-truth distribution, p (*Equation 1, Appendix*). As it relates to mutual information, KL-divergence and mutual information are intrinsically linked (*Equation 1* and *Equation 3*, *Appendix*). Crucially, the KL-divergence to mutual information relationship can be expressed as:

$$\textbf{(1)}\quad \boldsymbol{I(X;Y) = D_{KL}(p_{XY}||p_X p_Y)}$$

(*Equation 1: Entropy, Cross Entropy & KL Divergence, Appendix*)

Autoencoders consist of an encoder, a bottleneck (latent layer), and a decoder, where:

- The ***encoder*** is responsible for mapping data onto a low-dimensional space (i.e., latent layer).
- The ***latent layer*** (vector) represents a low-dimensional description of the data that can be manipulated (controlled) to enforce robust mappings.
- The ***decoder*** embeds the data in its original space from the learned description on the latent manifold. [2]

There are many methods for controlling the robustness of this mapping, including various latent distribution regularisation techniques, or adding noise to the input or latent variable:

$$\textbf{(2)}\quad \boldsymbol{z = y + \epsilon}\,,\ \textit{where…}$$

$\boldsymbol{z}$ is a noisy latent representation of the input

Balancing the Signal to Noise (SNR) ratio during training encourages a model to learn to remove this noise from the signal, ultimately leading to a more robust model.

Further, controlling the information rate through the latent layer can facilitate information flow during training (*Equation 2, Appendix*). In many autoencoder settings, we motivate Gaussianity to allow the maximum amount of information to be passed through latent neurons.[1] As a consequence of **(1)**, for a single Gaussian neuron, mutual information can be calculated directly from the variances:

$$\textbf{(3)}\quad \boldsymbol{I(Y;Z) = \frac{1}{2}\log\frac{\sigma_z^2}{\sigma_\epsilon^2}}$$

(*Equation 2: Information Rate (IR) Equation and IR for a Gaussian Neuron, Appendix*)

Autoencoders often use the MSE as an objective (loss) to determine how well a model can reconstruct input data from the latent distribution (*Equation 3*, *Appendix*). In this context, the MSE represents the average squared difference between the input and the reconstructed output across all input features. Further, minimising the MSE as a loss function encourages the autoencoder to produce reconstructed outputs that are as close to the original inputs as possible.

Tangentially, when we fix the errors to a normal distribution (zero-mean and fixed variance) and ensure that the residuals (reconstruction errors) follow a Gaussian distribution, the MSE can be interpreted probabilistically as minimising the negative log-likelihood (i.e., maximising the log-likelihood) of the data (*Equation 5*, *Appendix*). That is:

$$\textbf{(4)}\quad \boldsymbol{\arg min_\theta \sum (y_i - f(x_i;\theta))^2 \approx}$$
$$\boldsymbol{\theta^* = arg\ max_\theta \sum \log p(y_i|x_i,\theta)}$$

(*Equation 4: Probabilistic Formulation of the MSE, Appendix*)

The variational autoencoder (VAE) is a generative structure that extends this set-up by including a KL-divergence (D_{KL})

[1] Because Gaussian distributions are the most expressive.

term which encourages the latent variable distribution to take on the form of a prior (Gaussian) distribution. It can be further manipulated into an evidence lower bound (ELBO) for generative systems (*Equation 6, Appendix*). A performant model makes the encoder extraneous one training is complete, as similar data can be generated by sampling from the learned latent space and passing it through the decoder.

As it relates to the extended investigation, hyperparameter tuning can be done through annealing or Bayesian optimisation. Annealing is the process of gradually adjusting hyperparameters (or weights) during training. The use of β-KL (or λ-MMD) annealing can allow smooth changes to the D_{KL} values used for training over epochs. [3] In contrast, Bayesian optimisation uses past evaluation results to build a probabilistic model of the objective function and new hyperparameters are selected by optimising this model, focusing expressly on search areas more likely to yield good results. [4]

This context and background helps establish a precedent and motivation for subsequent investigations.

3. EXPERIMENT

For this set of experiments, a central model was gradually built upon to increase the complexity of investigation tasks.

The experiment began with a basic (deterministic) autoencoder in which Gaussian noise was added directly to the latent variable, and the latent space was "controlled" using noise variance and SNR without latent regularisation (*Table 1*, *Appendix*).

From a basic autoencoder, the next logical step was to encourage structure (non-VAE) in the latent space using Gaussian noise and a simple latent regularisation technique that penalised deviations of $\boldsymbol{z}$ mean and variance from zero and one, respectively. This phase of the experiment focused largely on the influence of the SNR, latent dimensions, and information flow (rate) (*Table 2*, *Appendix*).

The next iteration of the model introduced generative modeling, true stochasticity, and increased control over the latent space. This was done by incorporating a generative function, reparameterisation trick (*Figure 2*) and KL-divergences (*Equation 1*) used in VAEs (*Table 3*, *Appendix*).

The model then removed KL-divergence from the prior-fitting, resulting in a *VAE-like* structure, but using a kernel method (MMD/Gaussian kernel) to measure the difference between the latent space distribution and standard Gaussian distribution. This decision was motivated by the desire to demonstrate a non-standard VAE approach *and* by the knowledge that MMD is more flexible than a KL-divergence, as it can be used to control the latent space to follow any desired distribution (*Table 4*, *Appendix*).

The final phase of the experiment was a novel investigation that incorporated KL-divergence and MMD regularisers in the posterior-prior part of the model, to seek complimentary effects or an improved latent representation. Annealing and Bayesian optimisation were trialed as balancing approaches. For clarity, the following is an expression of the experiment:

$$\textbf{(5)}\quad \mathcal{L}_{\boldsymbol{total}} = \mathcal{L}_{\boldsymbol{reconstruction}} + (\boldsymbol{\beta} \cdot \mathcal{L}_{\boldsymbol{KL}} + \lambda \cdot \mathcal{L}_{\boldsymbol{MMD}})$$

With a simple annealing schedule, and later, Optuna.

$$\textbf{(6)}\quad \beta_t = min\left(1, \frac{t}{T_{anneal}}\right)$$

3.2. Results Discussion

Table 1 – Table 5 (*Appendix*) displays notable outcomes, model configurations, and limitations from the full investigation. The most performant autoencoder for data reconstruction was derived from model $\boldsymbol{f_2}$ (*Table 2*) at SNR 15 with 6 latent dimensions. This model allowed the autoencoder to learn a detailed and efficient representation of the data using a high enough SNR to capture information for accurate reconstruction (Total Loss = 0.0052) without increasing dimensionality unnecessarily or becoming prone to the overfitting and complexity increases observed at SNR 20.

Overall, the information rate steadily increased at higher SNRs (stabilising around 2.2 – 3.5 bits for SNR 15) and the model demonstrated clear reconstruction improvements compared to lower SNRs, such as 5 and 10, where noise overly-distorted the signal.

The model's generative performance did not attain quite the same success as its reconstructive performance. The best (qualitative) output came from model $\boldsymbol{f_4}$ (*Table 4*), which used a VAE-like structure with a Gaussian (kernel) MMD loss. The final reconstruction loss was 0.0042, the MMD loss was 0.0408, and the generative KL-divergence was 2.983.

This result led to a pivot in the investigation of the latent space and its distribution to determine if using both KL-divergence and MMD together in regularisation might further enhance the latent representation and lead to better generative outputs.

This hypothesis was proven false by model $\boldsymbol{f_5}$, in which it became clear that forcing the model to try to satisfy two different regularisation objectives for the same posterior-prior distribution only worsened the model's generative performance (as evidenced by increasing $\boldsymbol{D_{KL}}$ scores). Annealing and Bayesian optimisation further supported the null hypothesis by providing evidence that the performance decrease was not a result of beta or lambda settings.

4. CONCLUSION

In conclusion, this study explored the development of a non-variational autoencoder to reconstruct and generate 3D data with increasing complexity. The model succeeded in achieving high quality reconstructions at a handful of SNR and latent dimension settings, but arguably did so the most efficiently at an SNR of 15 with 6 latent dimensions. This proved the efficacy of controlling the latent space with Gaussian noise and regularisation techniques.

However, the inclusion of both KL-divergence and MMD regularisers within the posterior-prior proved to be counterproductive for generative tasks, and the application of annealing and Bayesian optimisation reaffirmed these findings.

A more positive direction for future research would be to deconstruct the posterior-prior regularisation into a single, preferred measure and target annealing or Bayesian optimisation at the two major ELBO terms to develop a β-VAE model with KL-annealing.

5. APPENDIX

5.1. Equations

Where ***entropy***, the average information of *Z,* is expressed as:

(7) $\boldsymbol{H(z) = H(p) = E_{z\sim p}[-\log p(z)]}$

The ***cross entropy*** of model *q* and the true distribution *p* is…

(8) $\boldsymbol{H(p,q) = E_{z\sim p}[-\log q(z)]}$

… for which we aim to find *q* that minimises *H(p,q)*, such that *q* is similar to *p*.

As it relates to neural networks, if $q_\theta$is characterized with parameters θ, then finding optimal parameters θ is equivalent to finding a variational approximation q_θ of *p*.

The ***Kullback-Leibler (KL) divergence*** is defined as:

(9) $\boldsymbol{H(p||q) = [H(p,q) - H(p)] = E_{z\sim p}\left[\log \frac{p(z)}{q(z)}\right]}$

(10) $\boldsymbol{KL(p,q) = 0}$ *when* $\boldsymbol{p = q}$

The natural objective for finding model *q* is minimal KL-divergence (i.e., minimising cross entropy as *H(p)* constant).

Crucially, the KL-divergence to mutual information relationship can be expressed as:

(11) $\boldsymbol{I(X;Y) = D_{KL}(p_{XY}||p_X p_Y)}$

… making everything that is true for the KL-divergence *also* true for mutual information (*Equation 2, below*).

Equation 1: Entropy, Cross Entropy & KL Divergence. *[5]*

The formula for mutual information is the Kullback-Leibler (KL) Divergence (i.e., the difference) between the joint distribution of Y and Z, and the multiplication of the marginal [2] (*Equation 1, above*).

(12) $\boldsymbol{I(Y;Z) = KL(p_{YZ}, p_Y p_Z) = \int p(y,z) \log \frac{p(y,z)}{p(y)p(z)} = E_{YZ} \log \frac{p(Z|Y)}{p(Z)}}$

For a single Gaussian neuron, mutual information can be calculated directly from the variances (That is, entropy for Gaussian distributions directly relates to their variance):

(13) $\boldsymbol{I(Y;Z) = \frac{1}{2} \log \frac{\sigma_Z^2}{\sigma_\epsilon^2}}$

Equation 2: Information Rate (IR) Equation and IR for a Gaussian Neuron. *[6]*

(14) $\boldsymbol{MSE = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{x}_i)^2}$

Equation 3: Mean Squared Error (MSE) Loss. *[5]*

Have a model that predicts an output *y*, based on input *x* such that (standard neural network):

(15) $y = f(x;\theta) + \epsilon$, *where…*

$f(x;\theta)$ is a model's prediction parameterised by θ, and ϵ is the noise.

In an autoencoder setting, we make noise ϵ follow a Gaussian (normal) distribution with zero-mean and variance σ^2:

(16) $\boldsymbol{\epsilon} \sim \mathcal{N}(0,\sigma^2)$ **,** *hence …*

(17) $\boldsymbol{y} \sim \mathcal{N}(f(x;\theta),\sigma^2)$

Under Gaussian assumption, likelihood of observing *(x,y)* is:

(18) $p(y|x;\theta) = \frac{\mathbf{1}}{\sqrt{\mathbf{2\pi\sigma^2}}}\ exp\left(-\frac{\boldsymbol{(y-f(x;\theta))^2}}{\mathbf{2\sigma^2}}\right)$

(19) $\log p(y|x;\theta) = -\frac{\mathbf{1}}{\mathbf{2}}\log(2\pi\sigma^2) - \frac{\boldsymbol{(y-f(x;\theta))^2}}{\mathbf{2\sigma^2}}$

… which makes estimating model parameters θ equivalent to maximising the likelihood of the data (i.e., minimising the negative log likelihood and/or minimising the squared difference between the prediction and ground truth).

Thus, the probabilistic formulation of the MSE can be expressed mathematically as,

(20) $\boldsymbol{arg\ min_\theta \sum (y_i - f(x_i;\theta))^2,}$

which [in this context] is equivalent to…

(21) $\boldsymbol{\theta^* = arg\ max_\theta \sum \log p(y_i|x_i,\theta)}$

Equation 4: Probabilistic Formulation of the MSE. *[5]*

Set out to build generative system: $\boldsymbol{p_\theta(x)}$

Probabilistic encoder provides latent *z* distribution given *x*:

$$\boldsymbol{q_\phi(z|x)}$$

Using Jensen inequality with concavity, can resolve the log likelihood to an **Evidence Lower Bound (ELBO)**:

(22) $\boldsymbol{E_{q_\phi(z|x)} \log p_\theta(x|z) - D(q_\phi(z|x)||p(z))}$

Which becomes a lower bound on the likelihood that is maximized over $\boldsymbol{\phi}$ (encoder parameters) and $\boldsymbol{\theta}$ (decoder parameters), instead of the likelihood itself.

ELBO represents:

- **reconstruction error (1st term) +**
  KL posterior Z to prior Z (2nd term)

In practice, these terms clash numerically.

Equation 5: VAE and ELBO. *[6]*

5.2. Figures & Tables

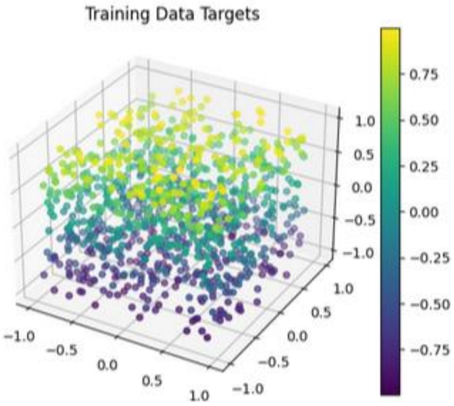


Figure 1: Training Data Targets.

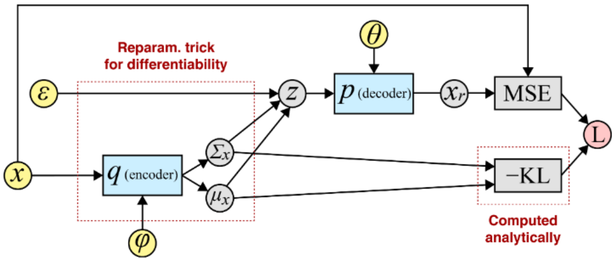
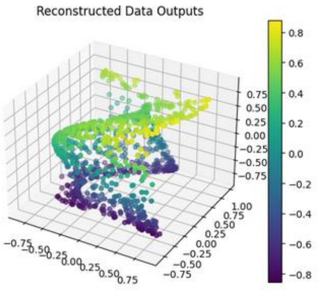


Figure 2: Reparameterisation Trick. *[2]*

***Model f_1*: Base Deterministic Model**

- **Encoder:** FCNN that maps input data to latent z.
- **Decoder:** FCNN that reconstructs input data from z.
- **Latent Space:** Gaussian noise added to z using SNR; noise is deterministic.
- **Loss:** MSE compares reconstruction.

Limitations: No explicit control over latent space properties; can have any distribution; no constraints or regularisation



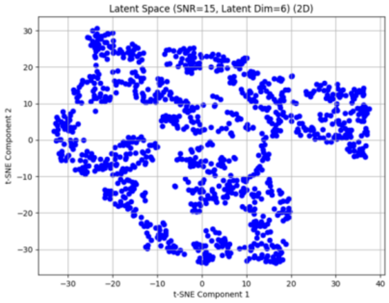
Final MSE Loss = 0.0920

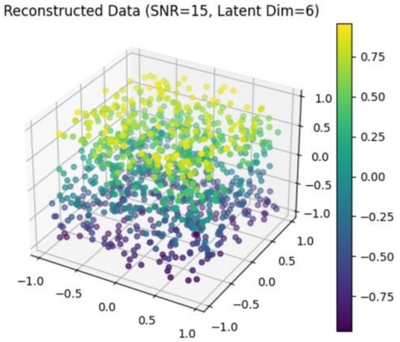
Table 1: Model f_1 Summary and Output.

***Model f_2*: Modified Base Model**

- **Encoder:** Same as $\boldsymbol{f_1}$
- **Decoder:** Same as $\boldsymbol{f_1}$
- **Latent Space:** Gaussian noise added to z using SNR; latent space is regularised to follow zero-mean, variance-one.
- **Information Rate:** Shows information passing through latent layer.
- **Loss:** Total loss = MSE + latent reg. loss

Limitations: Noise is still deterministic, as not sampling from a probabilistic distribution but yields a handful of well-reconstructed outputs. Useful for reconstruction, but not strong in a generative setting.





SNR=15 & Latent Dimensions = 6

```
Epoch 000 | L: 0.9352 | IR: 3.006 bits
Epoch 100 | L: 0.0095 | IR: 2.494 bits
Epoch 200 | L: 0.0071 | IR: 2.503 bits
Epoch 300 | L: 0.0064 | IR: 2.513 bits
Epoch 400 | L: 0.0062 | IR: 2.498 bits
Epoch 500 | L: 0.0055 | IR: 2.527 bits
Epoch 600 | L: 0.0053 | IR: 2.530 bits
Epoch 700 | L: 0.0052 | IR: 2.504 bits
Epoch 800 | L: 0.0053 | IR: 2.510 bits
Epoch 900 | L: 0.0052 | IR: 2.528 bits
```

Final Total Loss = 0.0052, IR: 2.528 bits

Table 2: Model f_2 Summary and Output.

Model f_3: Variational Autoencoder (VAE)

- **Encoder:** Encoder outputs mu and log variance; instead of directly using z.
- **Decoder:** Same as $\boldsymbol{f_1}$.
- **Latent Space:** Latent variable is sampled using reparameterised *Equation 2 & Figure 2*…

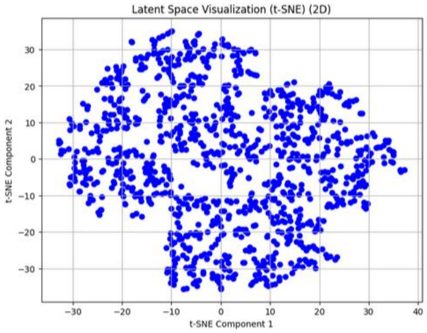
$$\boldsymbol{z} = \boldsymbol{y} + \boldsymbol{\epsilon} \cdot \sigma$$

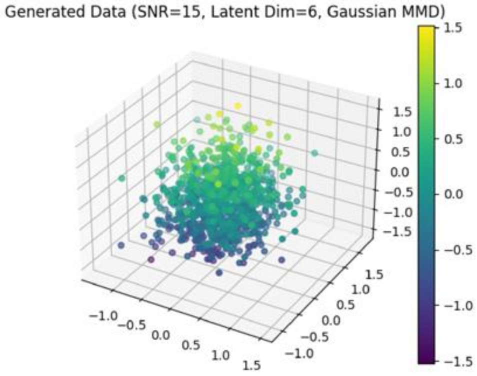
$$\text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(0,1) \text{ and } \sigma = \boldsymbol{exp}\left(\frac{\boldsymbol{log\ var}}{\mathbf{2}}\right)$$

  which introduces stochasticity into latent space for probabilistic sampling.

- **Loss:** Total loss = MSE + (KL-divergence b/w latent variable distribution and standard Gaussian distribution).
- KL-divergence regularises the latent space, ensuring latent distribution remains close to Std. Gaussian.

Limitations: True stochasticity and latent space regularisation, but sampling also increased variance in reconstructions.


Latent Space Visualization (t-SNE) (2D)


Generated Data (SNR=15, Latent Dim=6, Gaussian MMD)

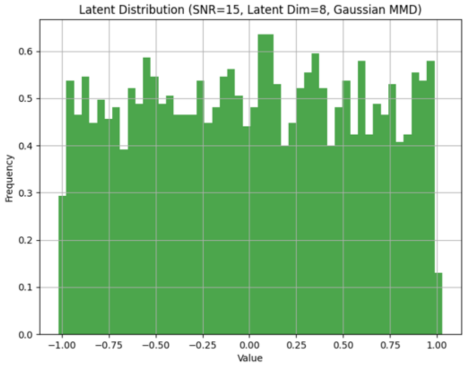
Final Total Loss = 0.3467

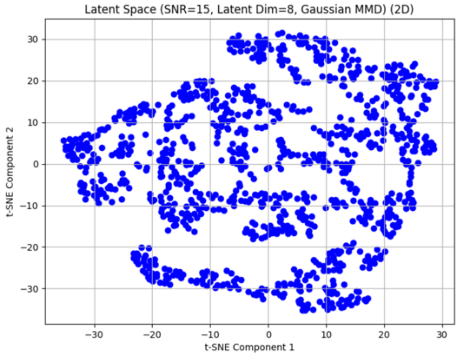
Table 3: Model f_3 Summary and Output.

Model f_4: VAE-like Model with MMD Regularisation

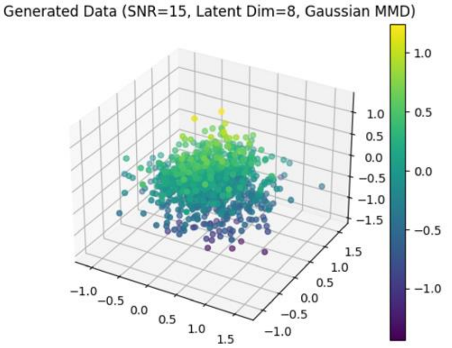
- **Encoder:** Same as VAE with reparameterisation.
- **Decoder:** Same as VAE with reparameterisation.
- **Latent Space:** Instead of KL-divergence, MMD was used to encourage latent space to follow Gaussian distribution (another model with identical set-up used Std. Gaussian). See *Figure B*, *Figure C Verbose Figures* section, if desired.
- **Loss:** Total loss = MSE + (MMD loss b/w latent variable distribution and Std./Gaussian distribution)
- MMD regularises the latent space, ensuring latent distribution remains close to Std./Gaussian.
- **Generative Performance:** Evaluated using D_{KL}

Limitations: Pairwise kernel evaluations are more expensive to compute and required additional consideration for optimal *r* in the Gaussian kernel (median heuristic). [1]


Latent Distribution (SNR=15, Latent Dim=8, Gaussian MMD)


Latent Space (SNR=15, Latent Dim=8, Gaussian MMD) (2D)

Cont'd.



Best (Qual) Generative Performance:
SNR 15 | Latent Dims = 8

Loss = 0.0042 | MMD loss = 0.0408

Generative D_{KL} = 2.983

Table 4: Model f_4 Summary and Output.

Model f_5: Novel Investigation - VAE-like Model; Two Latent Regularisation Terms

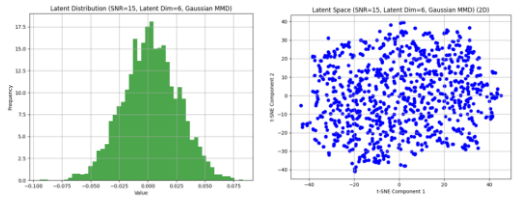
- **Encoder:** Same as VAE with Gaussian MMD.
- **Decoder:** Same as VAE with Gaussian MMD.
- **Latent Space:** Included both D_{KL} and MMD to Std. Gaussian distribution within the regularisation of the latent space.

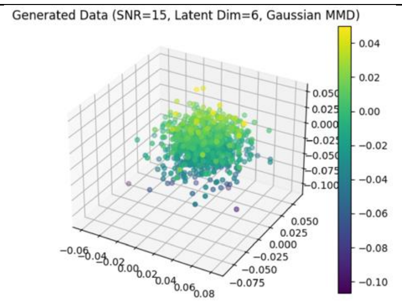
  Annealing schedules for KL-beta and MMD-weight were introduced to explicitly control the regularisation and of D_{KL}/MMD losses independently to observe effects on the latent space and overall model.
- **Loss:** Total loss = MSE (reconstruction loss) + annealed KL-divergence b/w latent variable distribution and standard Gaussian distribution + annealed MMD loss b/w latent variable distribution and standard Gaussian distribution).

Limitations: Both KL divergence and MMD aim to align the posterior distribution with the prior distribution but do so differently.

KL encourages minimizing the relative entropy between the posterior and prior, while MMD encourages minimizing the difference in means and variances.

Fixed Annealing





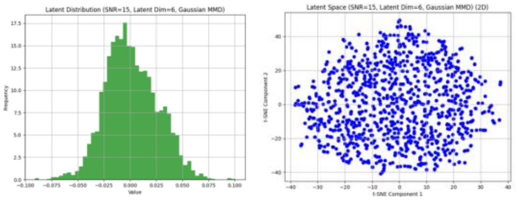
β = 1.0 → 10 λ = 0.01 → 1.0

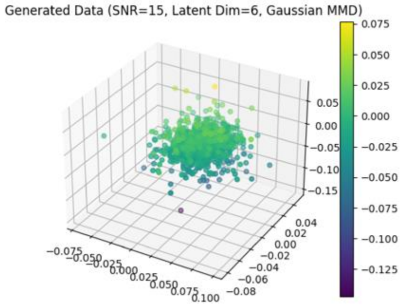
Testing with SNR = 15, Latent Dimensions = 6

```
Epoch 000 | L: 184.92 | MMD L: 0.0357
Epoch 100 | L: 1.2056 | MMD L: 0.0331
Epoch 200 | L: 0.7092 | MMD L: 0.0308
Epoch 300 | L: 0.5678 | MMD L: 0.0397
Epoch 400 | L: 0.5299 | MMD L: 0.0322
Epoch 500 | L: 0.5021 | MMD L: 0.0346
Epoch 600 | L: 0.4792 | MMD L: 0.0354
Epoch 700 | L: 0.4698 | MMD L: 0.0355
Epoch 800 | L: 0.4606 | MMD L: 0.0324
Epoch 900 | L: 0.4527 | MMD L: 0.0346
…
Epoch 1400 | L: 0.4364 | MMD L: 0.0309
```

Final KL-divergence = 4.517

Bayesian Optimisation (Optuna)





Testing with SNR = 15, Latent Dimensions = 6

```
Epoch 000 | L: 167.8 | KL-Beta: 1.0 | MMD-Weight: 0.01 | MMD Loss: 0.0294
Epoch 100 | L: 1.981 | KL-Beta: 1.9 | MMD-Weight: 0.02 | MMD Loss: 0.0318
Epoch 200 | L: 1.240 | KL-Beta: 2.8 | MMD-Weight: 0.03 | MMD Loss: 0.0306
…
Epoch 1000 | L: 0.7620 | KL-Beta: 10 | MMD-Weight: 0.1 | MMD Loss: 0.0278
…
Epoch 1400 | L: 0.6700 | KL-Beta: 10 | MMD-Weight: 0.1 | MMD Loss: 0.0271
```

Final KL-divergence = 7.050

Table 5: Model f_5 Summary and Output.

6. REFERENCES

[1] M. DaRocha, “Demonstrating the Basic Principles of Generative Models for (FCNN) Gaussian and Uniform Mappings,” *VUW, AIML425: Neural Networks and Deep Learning*, Aug. 2024, Accessed: Sep. 10, 2024. [Online]. Available: https://github.com/Marianette/A2-AIML425

[2] B. Kelijn, “Autoencoders: AIML425 [Lecture 7 Notes],” *Master of Artificial Intelligence, Victoria University of Wellington*, Sep. 2024.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Generative Models*. MIT Press, 2016. [Online]. Available: https://www.deeplearningbook.org/

[4] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, “Taking the Human Out of the Loop: A Review of Bayesian Optimization,” *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016, doi: 10.1109/JPROC.2015.2494218.

[5] B. Kleijn, “Brief Review Probability Theory and Basic Information Theoric Quantities as Used in Deep Learning: AIML425 [Lecture 1 Notes],” *Master of Artificial Intelligence, Victoria University of Wellington*, pp. 16–20, Jul. 2024, Accessed: Jul. 18, 2024. [Online]. Available: https://ecs.wgtn.ac.nz/foswiki/pub/Courses/AIML425_2024T2/LectureSchedule/probability.pdf

[6] B. Kleijn, “Autoencoders: AIML425 [Lecture 7 Notes],” *Master of Artificial Intelligence, Victoria University of Wellington*, Sep. 2024, Accessed: Sep. 22, 2024. [Online]. Available: https://ecs.wgtn.ac.nz/foswiki/pub/Courses/AIML425_2024T2/LectureSchedule/autoEnc.pdf

7. VERBOSE FIGURES

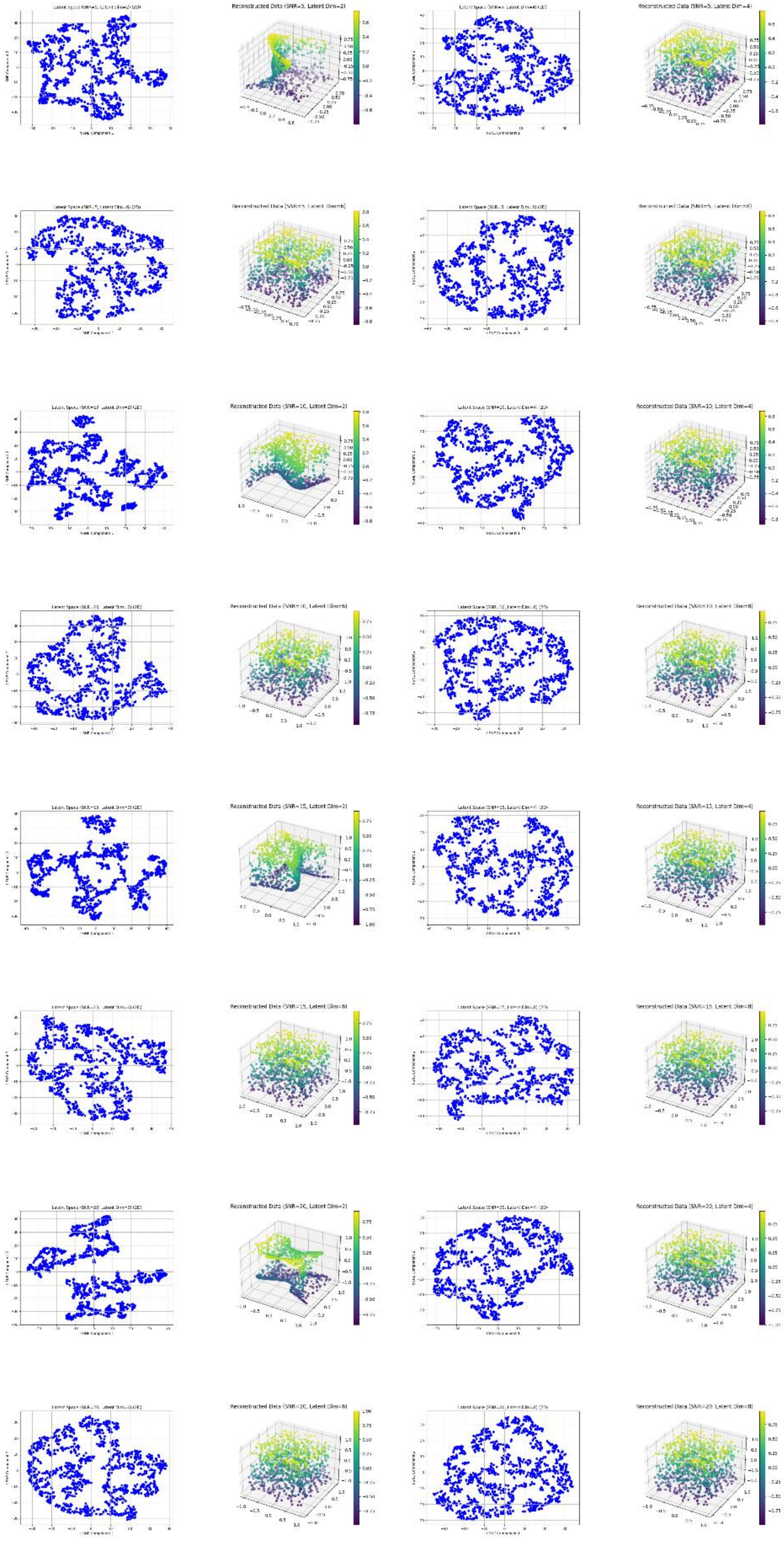


Figure A: f_2 SNR and Latent Dim Reconstructions

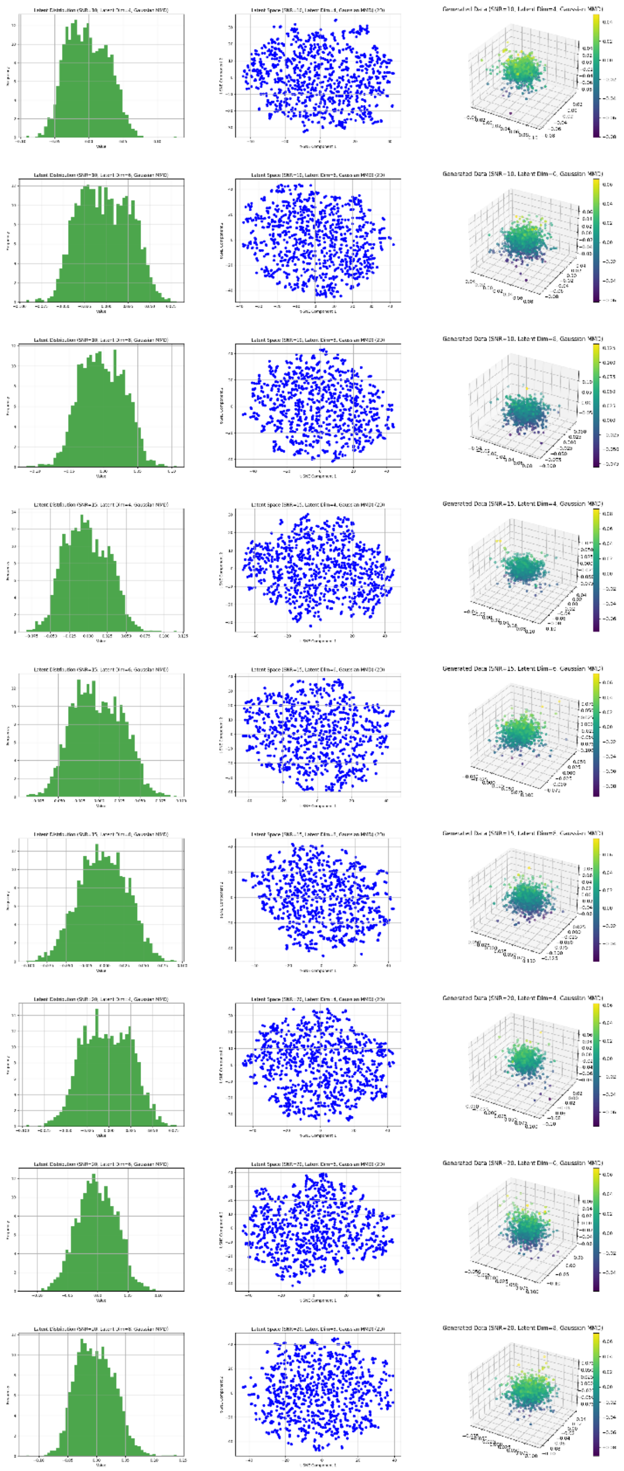


Figure B: f_4 VAE-like MMD loss, Std. Gaussian

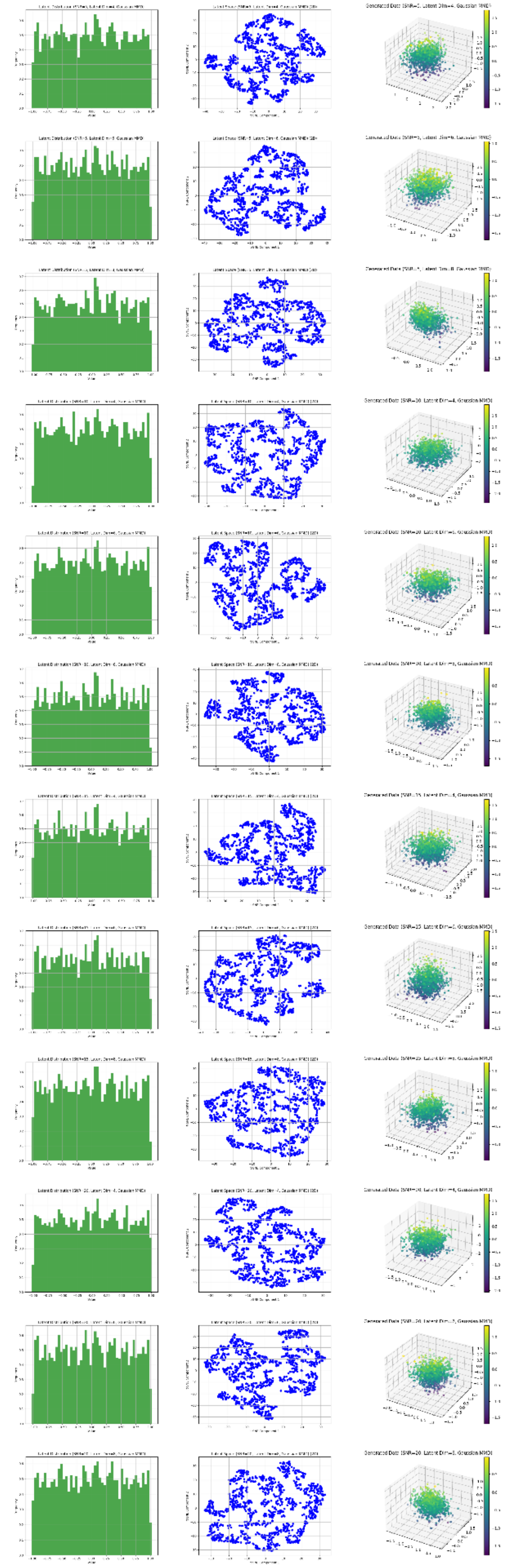


Figure C: f_4 VAE-like MMD loss, Gaussian