Embedding Edit Distance with Variational Autoencoders

Chien-An Chou[1]

[1]Department of Electrophysics, Department of Electrical Engineering, National Yang Ming Chiao Tung University, Hsinchu 300, Taiwan

June 2, 2025

Abstract

This challenge explores how variational autoencoders (VAEs) can learn latent representations that capture string similarity, specifically edit distances between binary sequences. We integrate edit distance supervision, KL regularization, and binary reconstruction loss to analyze their roles via ablation studies, correlation evaluation, and visualization. Key metrics include Pearson correlation and reconstruction fidelity.

1 Introduction

Edit distance is a widely used metric to quantify the similarity between strings by counting the minimum number of operations (insertions, deletions, substitutions) required to transform one string into another. This concept is central in various domains such as bioinformatics, natural language processing, and information retrieval.

Despite its interpretability and effectiveness, computing edit distance can be computationally expensive and non-differentiable, posing challenges for integration into gradient-based deep learning frameworks. In recent years, deep generative models like Variational Autoencoders (VAEs) have shown promise in learning meaningful representations from data.

In this challenge, we explore whether it is possible to embed binary string sequences into a continuous latent space where Euclidean distances approximate edit distances. Our hypothesis is that an effective encoder-decoder architecture, trained with appropriate regularization, can learn a latent structure where similar sequences lie closer together.

To achieve this, we design and train a VAE with an additional loss term that aligns latent distances with ground-truth edit distances. This enables us to investigate how each loss componentreconstruction error (BCE), KL divergence, and edit losscontributes to the latent space geometry and reconstruction fidelity.

Our experiments are designed to answer the following questions:

- Can VAEs learn latent embeddings that correlate with edit distance?
- How do different regularization strategies affect the Gaussianity and interpretability of the latent space?
- What is the trade-off between reconstruction accuracy and edit distance alignment?

2 Data Generation

To evaluate whether neural models can embed edit similarity, we generated synthetic binary sequence datasets with controlled properties. Specifically, we created two types of datasets:

- **Variable-length sequences**: Each sample is a binary string with length uniformly sampled between 20 and 100 bits. This tests the models robustness to varying input lengths.
- **Fixed-length sequences**: All samples are 100-bit binary strings, allowing for stable training and easier model comparison.

For each dataset, we computed the pairwise edit distances between all sequence pairs using the standard Levenshtein distance. These distances were normalized to the range $[0, 1]$ and stored in precomputed matrices. During training, these matrices serve as ground-truth supervision for structuring the latent space: ideally, sequences that are close in edit distance should also be close in the learned latent space.

This setup provides a principled benchmark for assessing whether the learned embeddings preserve discrete sequence similarity in a continuous space.

Visualization and Observations.
To better understand the characteristics of our synthetic datasets, we visualized the distributions of sequence lengths and pairwise edit distances across varying sizes (50, 200, 500) and average edit difficulty levels (2, 5, 10).

We observe that:

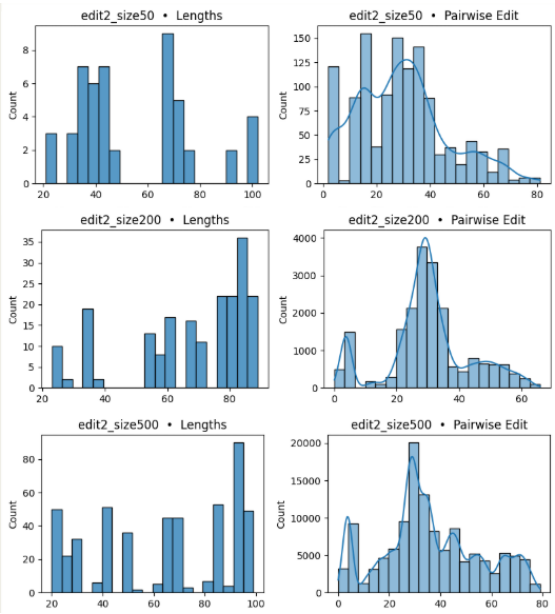
- The sequence length distributions are nearly uniform due to our sampling strategy, with sequences randomly chosen between 20 and 100 bits.

- The pairwise edit distance histograms tend to exhibit multi-modal or skewed shapes when sample sizes are small (e.g., $N = 50$), and gradually smooth out as N increases.

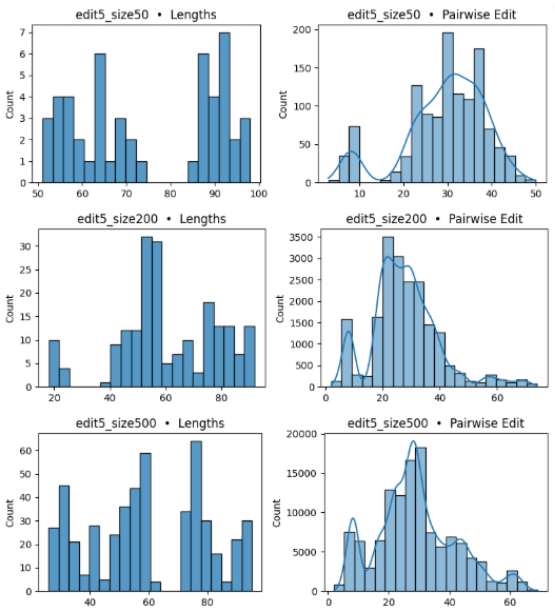
- However, due to computational limits, we only computed full pairwise distance matrices for datasets up to size 500. This limited the ability to observe a clear normal distribution shape, which would be expected in larger datasets due to the Central Limit Theorem.

Expected Distribution of Edit Distances.
In theory, when sampling many random binary strings and applying independent random bit flips, the pairwise edit distances between sequence variants should approximate a normal distribution centered around the expected number of edits. This arises from the additive nature of random mutations across positions, especially when the mutation process follows a fixed probability distribution.

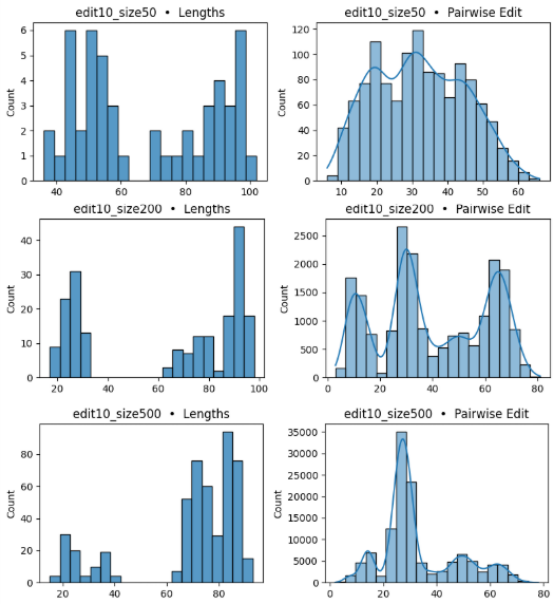
Dataset Visualization Edit Distance 2



Dataset Visualization Edit Distance 5



Dataset Visualization Edit Distance 10



3 Model Architecture

Our architecture is based on a standard Variational Autoencoder (VAE) adapted for binary sequence inputs. The model consists of three main components:

- **Encoder:** The input binary sequence is first reshaped into a 1D format and passed through two convolutional layers with ReLU activations. These layers capture local bit patterns. The output is flattened and projected through a dense layer to produce a 128-dimensional hidden representation.
- **Latent Space:** From the hidden representation, two separate linear layers compute the latent mean μ and log-variance $\log \sigma^2$. Latent vectors z are then sampled using the reparameterization trick:

$$z = \mu + \epsilon \cdot \sigma, \quad \text{where } \epsilon \sim \mathcal{N}(0, I)$$

- **Decoder:** The latent vector z is passed through a multilayer perceptron (MLP) with ReLU activation and a final sigmoid layer to reconstruct the original input bitwise.

This design allows us to encode binary sequences of arbitrary length into a compact latent representation, and decode them back while optimizing for bitwise accuracy and structural alignment.

```
class VAE(nn.Module):
    def __init__(self, input_length, latent_dim=16):
        super(VAE, self).__init__()
        self.encoder = nn.Sequential(
            nn.Conv1d(1, 32, kernel_size=3, padding=1)
    ,
            nn.ReLU(),
            nn.Conv1d(32, 64, kernel_size=3, padding
    =1),
            nn.ReLU(),
            nn.Flatten(),
            nn.Linear(64 * input_length, 128),
            nn.ReLU()
        )
        self.fc_mu = nn.Linear(128, latent_dim)
        self.fc_logvar = nn.Linear(128, latent_dim)
        self.decoder_fc = nn.Sequential(
            nn.Linear(latent_dim, 128),
            nn.ReLU(),
            nn.Linear(128, input_length)
        )
    def encode(self, x):
        h = self.encoder(x)
        return self.fc_mu(h), self.fc_logvar(h)
    def reparameterize(self, mu, logvar):
        std = torch.exp(0.5 * logvar)
        eps = torch.randn_like(std)
        return mu + eps * std
    def decode(self, z):
        return torch.sigmoid(self.decoder_fc(z)).
    unsqueeze(1)
    def forward(self, x):
        mu, logvar = self.encode(x)
        z = self.reparameterize(mu, logvar)
        recon_x = self.decode(z)
        return recon_x, mu, logvar, z
```

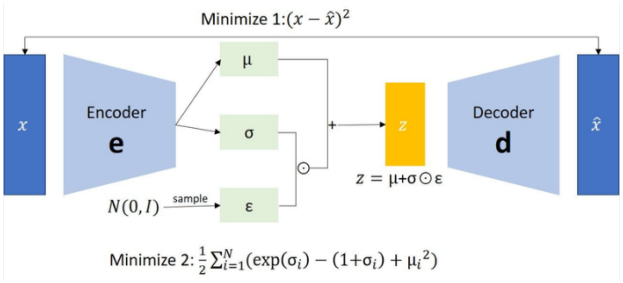


Figure 1:
Schematic of the VAE architecture with reparameterization and two loss objectives: reconstruction loss and KL divergence.

4 Loss Function

The training objective combines three components to simultaneously ensure accurate reconstruction, regularized latent space, and alignment with edit distance supervision:

$$\mathcal{L} = \mathcal{L}_{\text{BCE}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_{\text{edit}}\mathcal{L}_{\text{edit}}$$

- **Reconstruction Loss** ($\mathcal{L}_{\text{BCE}}$): Binary cross-entropy loss measures the bitwise difference between the input sequence x and the reconstruction $\hat{x}$. This term ensures that the decoder can faithfully reconstruct the original input from the latent vector z.

$$\mathcal{L}_{\text{BCE}} = -\sum_{i=1}^{L} [x_i \log \hat{x}_i + (1 - x_i) \log(1 - \hat{x}_i)]$$

- **KL Divergence** ($\mathcal{L}_{\text{KL}}$): This regularizes the latent space by encouraging the approximate posterior $q(z|x)$ to match a standard Gaussian prior $p(z) = \mathcal{N}(0, I)$. It enables meaningful interpolation and sampling in the latent space.

$$\mathcal{L}_{\text{KL}} = -\frac{1}{2}\sum_{j=1}^{d} \left(1 + \log \sigma_j^2 - \mu_j^2 - \sigma_j^2\right)$$

- **Edit Distance Supervision** ($\mathcal{L}_{\text{edit}}$): To encourage the latent embeddings to reflect discrete sequence similarity, we introduce a loss that matches the pairwise L2 distances between latent vectors with their normalized edit distances. This is implemented via mean squared error (MSE):

$$\mathcal{L}_{\text{edit}} = \frac{1}{N^2}\sum_{i,j} \left(\|z_i - z_j\|_2 - \text{Edit}(x_i, x_j)\right)^2$$

The weighting coefficients λ_{KL} and λ_{edit} control the trade-off between reconstruction quality, latent regularity, and semantic alignment with sequence structure.

5 Task 1: Embedding Edit Distance

A core objective of this challenge is to evaluate whether the latent space learned by a variational autoencoder reflects discrete edit similarity between sequences. To assess this, we perform both quantitative and qualitative analysis.

Pearson Correlation Coefficient

We use the Pearson correlation coefficient ρ to measure the alignment between edit distances and the corresponding L2 distances in latent space:

$$\rho = \frac{\mathrm{Cov}(D_{\text{edit}}, D_{\text{latent}})}{\sigma_{\text{edit}} \cdot \sigma_{\text{latent}}}$$

where D_{edit} denotes the ground truth pairwise edit distance matrix, and D_{latent} the corresponding latent space distances. A higher correlation indicates that the autoencoder has effectively embedded edit similarity into the latent representation.

Results and Comparison

Figure 1 shows a comparison of correlation values between trained and untrained models across different datasets. As expected, untrained models exhibit near-zero correlation, while trained models yield significantly higher alignment (up to 0.85), particularly for larger datasets.

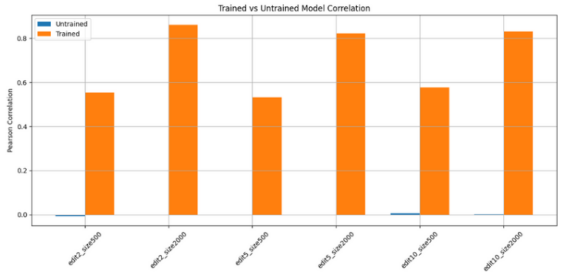


Figure 2: Pearson correlation between edit distance and latent L2 distance for trained and untrained models.

Figure 2 presents scatter plots of L2 versus edit distance before and after training, based on a small dataset ($N = 100$). The trained model shows a clear linear trend, demonstrating successful embedding of structural similarity.

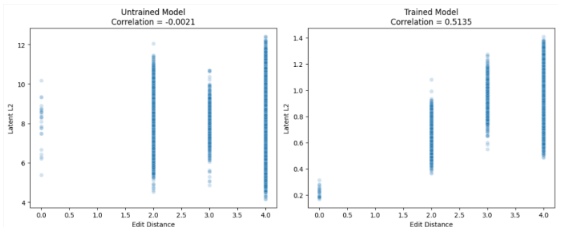


Figure 3: Scatter plots of edit distance vs. latent L2 distance. Left: untrained model; Right: trained model.

Visualization via t-SNE

To qualitatively assess the latent space geometry, we use t-SNE to project high-dimensional latent vectors into 2D. Figure 3 shows the t-SNE results for datasets of different sizes and edit difficulty levels. We observe more clustered and structured embeddings as dataset size increases, indicating improved learning of edit-based similarity.

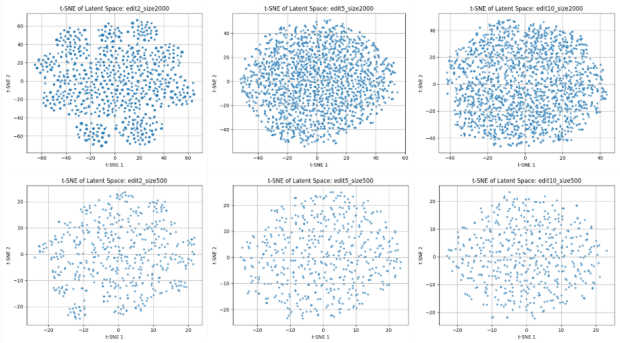


Figure 4: t-SNE visualization of latent vectors under different dataset sizes (rows) and edit difficulty levels (columns).

6 Task 2: Regularization Experiments

In this task, we study how the KL regularization weight λ_{KL} affects the structure of the latent space. Specifically, we explore whether the learned latent representations resemble samples from a standard normal distribution $\mathcal{N}(0, I)$.

Background. In the VAE framework, the KL divergence term encourages the posterior distribution $q(z|x)$ to match the standard normal prior. However, tuning the weight λ_{KL} is crucial:

- If λ_{KL} is too small, the model ignores the prior, and latent variables deviate from Gaussianity.
- If λ_{KL} is too large, the latent space collapses, and reconstruction quality degrades.
- A moderate λ_{KL} balances regularization and expressive capacity.

Setup. We train VAE models under $\lambda_{\text{KL}} \in \{0.0, 0.01, 0.1, 1.0\}$ and evaluate the resulting latent distributions using:

- **Histograms** of latent values (1D marginals) to visualize bell-shaped distributions.
- **Q-Q plots** comparing sample quantiles to the theoretical quantiles of a standard Gaussian.

Results. The figure below shows how the latent space evolves as λ_{KL} increases.

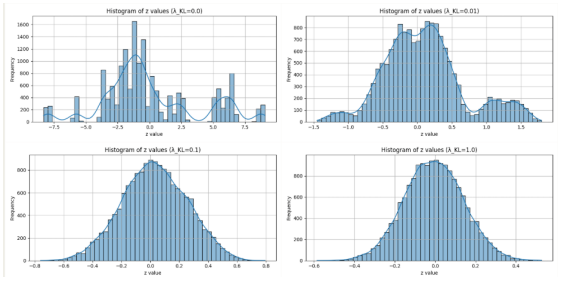


Figure 5: *Histogram observation:* At $\lambda_{\text{KL}} = 0.0$, the latent distribution is far from Gaussian. As λ_{KL} increases, the distribution becomes smoother and more centered.

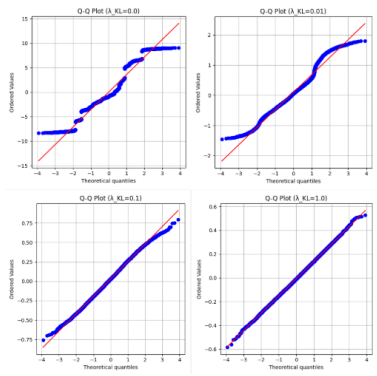


Figure 6: *Q-Q plot observation:* The points align more closely with the diagonal line as λ_{KL} increases, indicating increasing Gaussianity of the latent space.

Conclusion. These results confirm that λ_{KL} effectively controls the degree of regularization. A moderate value like 0.1 provides a good trade-off, making the latent space resemble $\mathcal{N}(0, I)$ while preserving useful encoding information.

7 Task 3: Reconstruction Fidelity

This task evaluates how well the model reconstructs binary sequences from the latent space. We consider two main reconstruction metrics:

- **Binary Cross Entropy (BCE):** Measures bitwise reconstruction quality.
- **Hamming Distance:** Quantifies total bit mismatch between original and reconstructed sequences.

Error vs. Sequence Length. We investigate whether reconstruction errors are sensitive to sequence length. Figure below shows the scatter plot of BCE against sequence length under different datasets.

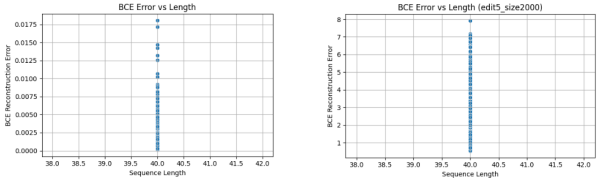


Figure: **BCE vs. Sequence Length.** Left: easy dataset (low edit difficulty). Right: hard dataset (edit5). The BCE remains low and stable across lengths, except under hard conditions where error increases.

Error vs. Edit Difficulty. We further study whether sequences that are harder (i.e., farther from others in edit distance) are reconstructed worse. We compare Hamming distance against average edit difficulty.

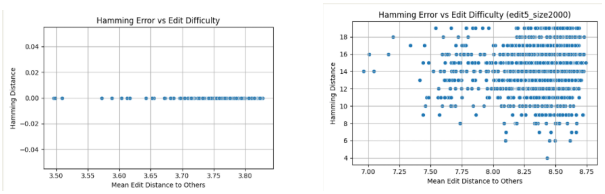


Figure: **Hamming Distance vs. Edit Difficulty.** Left: easy dataset. Right: edit5. Under harder settings, reconstruction fails more frequently as the edit difficulty rises.

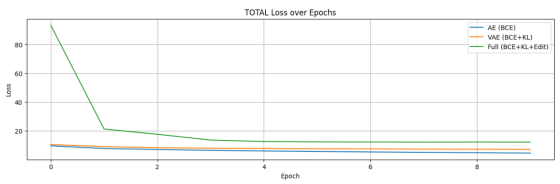
Takeaway. In summary, under challenging datasets (e.g., edit5), the models reconstruction performance degrades both with sequence difficulty and noise. These trends are minimal in easier datasets, indicating robustness only under mild edit regimes.

8 Task 4: Ablation Studies

To understand the role of each loss component in learning meaningful representations, we perform ablation experiments by selectively removing individual loss terms from the full objective:

- **No_KL:** Remove KL regularization. This allows the latent space to drift from $\mathcal{N}(0, I)$, potentially improving alignment with edit distance.
- **No_Edit:** Remove edit supervision. This prevents explicit optimization for matching edit distance in latent space.
- **No_BCE:** Remove reconstruction loss. This tests if edit supervision alone can maintain useful representations.

We plot the total and component losses over training epochs for each variant. Additionally, we track the Pearson correlation between latent L2 distances and edit distances to evaluate alignment.



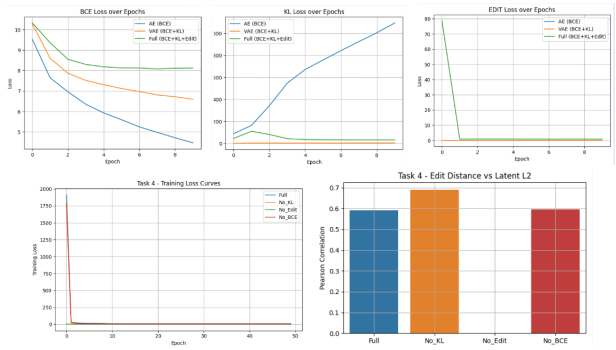


Figure: Loss breakdowns (top and middle rows) and final correlation results (bottom). Removing edit loss destroys alignment (No_Edit), while removing KL improves it (No_KL). Best performance occurs with BCE + Edit loss only.

Conclusions:

- **Edit loss is essential.** No_Edit fails to learn edit-aligned latent geometry.
- **KL may hurt alignment.** No_KL achieves the best correlation.
- **BCE helps reconstructions.** Omitting BCE weakens output quality but preserves embeddings.
- **Best tradeoff:** BCE + Edit (No_KL setup) balances alignment and reconstruction.

9 Conclusion

In this project, we investigated how well a Variational Autoencoder (VAE) can embed binary sequences such that latent L2 distances reflect discrete edit distances. Through a series of targeted experiments, we draw the following conclusions:

- **Edit supervision is essential.**
  Adding an explicit edit loss significantly improves the alignment between latent distances and ground-truth edit distances. Without it, the model fails to preserve structural similarity.
- **KL regularization may hurt alignment.**
  While KL divergence encourages a Gaussian latent prior, it can disrupt edit-distance preservation. Removing the KL term (`No_KL`) often yields better correlation with edit distances.
- **Reconstruction loss (BCE) ensures output fidelity.**
  Although omitting BCE does not destroy latent structure, it leads to degraded reconstruction quality. Including BCE helps the model produce bitwise-accurate outputs.
- **Latent space Gaussianity is controllable via $\lambda_{\mathbf{KL}}$.**
  Increasing KL weight produces smoother, bell-shaped distributions in latent dimensions, as confirmed by histograms and QQ plots. However, excessive KL strength risks posterior collapse.
- **Trade-offs exist between reconstruction and alignment.**
  The best embedding performance arises when combining BCE and edit loss, with minimal or no KL regularization. This setup balances structure preservation with accurate decoding.
- **Synthetic datasets validate embedding behavior.**
  Our variable-length synthetic binary datasets with controlled edit difficulty serve as effective testbeds for evaluating structural representation learning.