Poetry VAE: A Variational Autoencoder for Diverse Poetry Generation

Laiba Batool and Andleeb Zahra
Department of Data Science
FAST NUCES
Islamabad, Pakistan
i211781@nu.edu.pk, i212741@nu.edu.pk

***Abstract*—This paper presents Poetry VAE, a novel application of Variational Autoencoders (VAEs) for the generation of diverse poetry. While natural language generation has seen significant advances through various deep learning approaches, the generation of creative content like poetry presents unique challenges due to the complex interplay of linguistic structure, semantic meaning, and stylistic elements. We implement a VAE architecture enhanced with self-attention mechanisms and train it on the Poetry Foundation dataset containing diverse styles, themes, and structures. Our model maps poems into a continuous latent space, enabling both generation of new poems and controlled manipulation of poetic attributes. Through extensive experimentation and analysis, we demonstrate that VAEs can capture meaningful representations of poetic themes and structures, though challenges remain in generating consistently coherent output. We provide detailed visualizations of the learned latent space, analyze attention patterns, and present both quantitative metrics and qualitative samples of generated poetry at varying degrees of randomness. Our findings contribute to the ongoing research in creative text generation and offer insights into the capabilities and limitations of VAEs for poetry generation.**

***Index Terms*—Variational autoencoder, natural language generation, poetry generation, deep learning, self-attention mechanism, computational creativity**

I. Introduction

Poetry, with its nuanced expression, emotional depth, and structural complexity, represents one of the most sophisticated forms of human linguistic creativity. The computational generation of poetry presents a significant challenge for artificial intelligence systems, requiring not only grammatical correctness but also an understanding of meter, rhythm, thematic coherence, and emotional resonance.

Recent advances in deep learning have demonstrated impressive capabilities in natural language generation tasks, from dialogue systems to story generation. Among these approaches, Variational Autoencoders (VAEs) have emerged as a powerful framework for generating creative content due to their ability to learn continuous latent representations that can be sampled and manipulated.

In this paper, we explore the application of VAEs to poetry generation, with a specific focus on creating a model capable of generating diverse and stylistically varied poems. Our contributions include:

1) The development of a VAE architecture specifically tailored for poetry generation, incorporating bidirectional LSTM encoders, self-attention mechanisms, and techniques to improve generated text coherence
2) A comprehensive analysis of the learned latent space representing poetic content, demonstrating how semantic and stylistic features are organized
3) An investigation of techniques to address common challenges in VAE text generation, including KL divergence collapse and repetition in generated text
4) Qualitative and quantitative evaluation of generated poems across different sampling parameters
5) Visualization and analysis of attention patterns in the encoding of poetry

Our work builds on previous research in computational creativity and neural text generation but focuses specifically on the unique challenges and opportunities presented by poetic text. By examining both the technical aspects of the model architecture and the qualitative properties of the generated content, we aim to contribute to the broader understanding of how deep learning systems can generate creative content.

II. Related Work

The computational generation of poetry has a rich history, evolving from rule-based systems to sophisticated neural approaches. Early systems typically relied on templates, linguistic constraints, and hand-crafted rules to produce poetic text [1]. With the advent of deep learning, approaches have shifted toward data-driven models that can learn patterns from large corpora of poems.

A. Neural Approaches to Poetry Generation

Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, were among the first neural architectures applied to poetry generation [3] demonstrated that conditional variational autoencoders with hybrid decoders could generate thematic Chinese poetry with structure and coherence. Subsequent work explored various enhancements, including attention mechanisms [1] and hierarchical structures to capture both word-level and line-level patterns.

More recently, transformer-based models have shown impressive capabilities in generating coherent and stylistically diverse text. Particularly, large language models based on the transformer architecture have demonstrated remarkable abilities to generate poetry when fine-tuned on poetic corpora.

B. Variational Autoencoders for Text Generation

Variational Autoencoders have become an important tool for generative modeling across various domains. Unlike standard autoencoders, VAEs learn a probability distribution over the latent space, enabling more controlled and diverse generation.

Applying VAEs to text generation presents specific challenges due to the discrete nature of text and the sequential dependencies in language. Pioneering work demonstrated that these models could generate coherent sentences and enable smooth interpolation between different sentences in the latent space.

Several researchers have proposed modifications to improve VAE performance on text. The issue of ”KL divergence collapse,” where the model ignores the latent variable and defaults to a standard language model, has been addressed through techniques such as KL cost annealing, alternative training objectives, and architectural modifications.

C. Computational Creativity and Poetry

The field of computational creativity examines how computational systems can exhibit behaviors that would be deemed creative if performed by humans. Poetry generation represents a particularly interesting domain for computational creativity due to the complex interplay of constraints, meaning, and aesthetics.

Researchers have proposed that poetry generation systems should not only produce grammatically correct text but also respect poetic features like meter and rhyme while conveying coherent content. A comprehensive survey of computational approaches to poetry generation [1] highlights the evolution from rule-based systems to more sophisticated learning-based approaches.

Recent work has focused on how generative models can be tuned to produce more creative and surprising outputs while maintaining coherence. Methodologies for evaluating computational poetry have also evolved, moving beyond simple metrics to include human judgments of creativity, novelty, and value.

D. Attention Mechanisms and Self-Attention

Attention mechanisms, which allow models to focus on relevant parts of input sequences when producing outputs, have significantly improved performance in natural language processing tasks. Self-attention, where a sequence attends to itself, has proven particularly effective for capturing long-range dependencies and relationships between elements in a sequence.

In poetry generation, attention mechanisms can help the model focus on thematically or structurally important words and patterns [4] demonstrated that incorporating topical constraints through finite-state acceptors could improve the coherence and thematic consistency of generated poems, while their interactive poetry generation system Hafez [5] showed how these techniques could be applied in a user-friendly interface.

Our work builds on these foundations, combining VAEs with self-attention mechanisms specifically tailored for poetry generation, with a focus on analyzing the learned representations and the quality of generated poems.

III. Methodology

A. Dataset

We utilized the Poetry Foundation dataset from Kaggle, which contains a diverse collection of poems from different time periods, styles, and themes. The dataset includes approximately 13,800 poems along with metadata such as poet name, poem title, and thematic tags.

After preprocessing, we retained 5,591 poems that met our length and quality criteria. These poems were split into training (3,577 poems, 64%), validation (895 poems, 16%), and test sets (1,119 poems, 20%).

B. Data Preprocessing

Our preprocessing pipeline included several steps to prepare the raw text for training:

1) Text cleaning: We removed special characters while preserving essential punctuation markers (periods, commas, semicolons, etc.) that can indicate poetic rhythm and structure.
2) Tokenization: Using NLTK's word tokenizer, we converted each poem into a sequence of tokens.
3) Vocabulary construction: We built a vocabulary from tokens appearing at least 3 times in the corpus, resulting in a vocabulary size of 39,726 tokens. This included special tokens (<PAD>, <SOS>, <EOS>, and <UNK>) for padding, sequence start, sequence end, and unknown words, respectively.
4) Length filtering: We retained poems with lengths between 1 and 148 tokens (excluding start and end tokens) to ensure computational feasibility while preserving poem integrity.

The resulting tokenized poems were then converted to numerical sequences using the vocabulary mapping, with each poem prepended with a start token and appended with an end token.

C. Model Architecture

Our Poetry VAE model integrates several components designed specifically for poetry generation. The overall architecture is illustrated in Fig. 6.

1) Embedding Layer: The first component is an embedding layer that maps token indices to dense vector representations. We used an embedding dimension of 256, which provides sufficient expressivity while remaining computationally efficient.

2) Encoder: The encoder consists of a bidirectional LSTM network that processes the embedded poem sequences and captures contextual information in both forward and backward directions. We used a hidden dimension of 512 and 2 layers for the encoder LSTM, with dropout (0.5) applied to prevent overfitting.

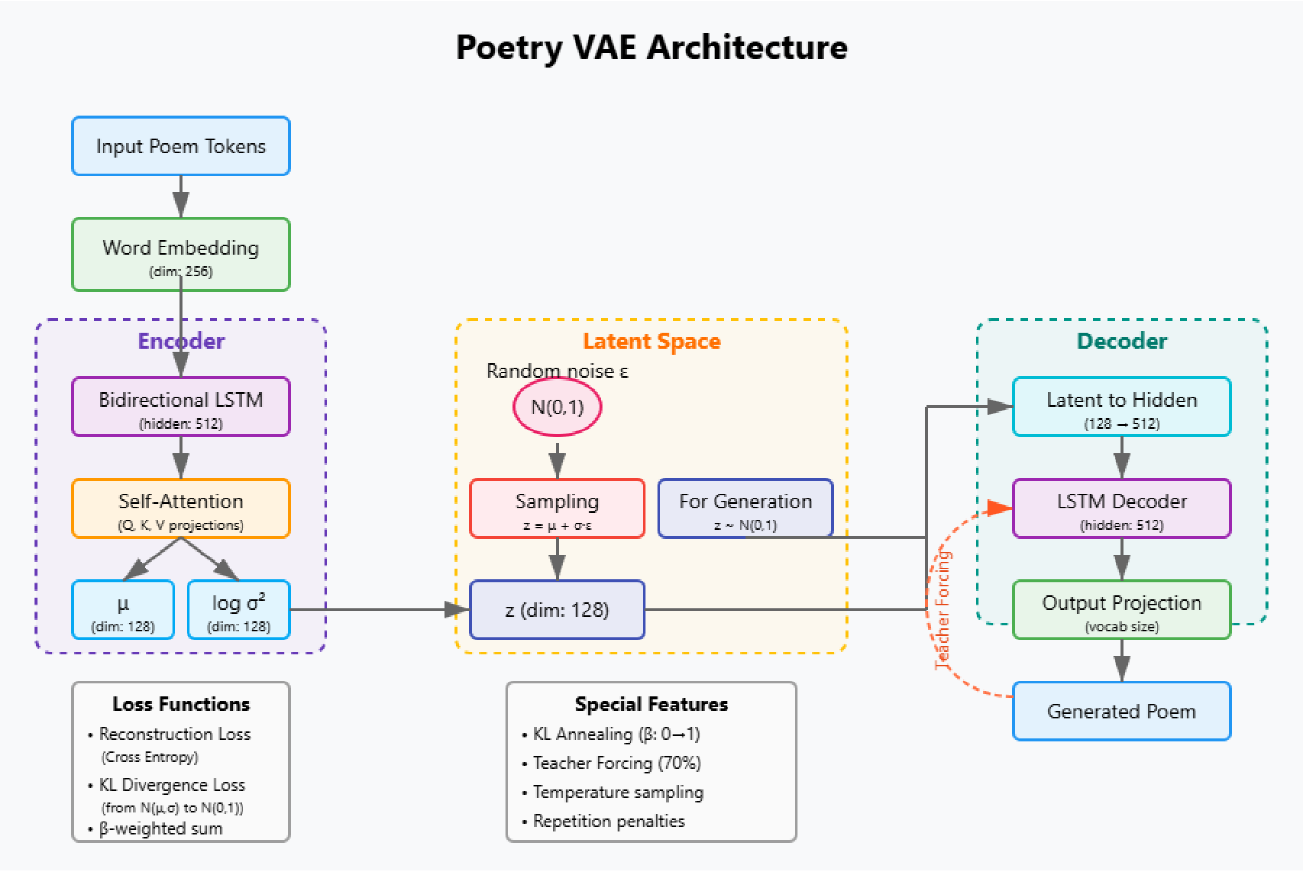

Fig. 1. Architecture of the Poetry VAE model. The encoder (left) processes input poems through embedded representation, bidirectional LSTM layers, and self-attention to produce parameters of the latent distribution. The decoder (right) reconstructs poems from samples in the latent space. Key components include embedding layers, LSTM networks, attention mechanism, and the sampling operation in the latent space.

A critical enhancement to our encoder is the self-attention mechanism, which allows the model to weigh the importance of different tokens when constructing the latent representation. The self-attention module computes query, key, and value projections of the LSTM outputs, producing attention weights that highlight the tokens most relevant to the overall meaning and structure of the poem.

The encoder outputs are then transformed through linear projections to predict the parameters (mean μ and log variance $\log \sigma^2$) of the latent distribution.

3) Latent Space: We designed a 128-dimensional latent space to capture the semantic and stylistic features of poems. During training, we sample from the distribution $\mathcal{N}(\mu, \sigma^2)$ using the reparameterization trick, which allows gradient flow through the sampling operation. The latent space serves as a compressed representation of the poem, encoding both semantic content and stylistic features.

4) Decoder: The decoder receives the sampled latent vector and reconstructs the original poem. It consists of an LSTM network with the same hidden dimension (512) and number of layers (2) as the encoder.

A key feature of our decoder is the concatenation of the latent vector with each input token embedding, ensuring the latent information influences each step of the decoding process. The decoder outputs are projected to the vocabulary space through a linear layer, producing token probabilities for each position in the sequence.

During training, we employed teacher forcing with a probability of 0.7, providing the model with the ground truth token instead of its own prediction at each decoding step.

D. Training Procedure

1) Loss Function: The VAE training objective consists of two components:

1) Reconstruction loss: Cross-entropy loss between the predicted token distributions and the actual tokens in the poem.
2) KL divergence loss: Kullback-Leibler divergence between the learned latent distribution $\mathcal{N}(\mu, \sigma^2)$ and the prior distribution $\mathcal{N}(0, 1)$.

The total loss is computed as:

$$\mathcal{L} = \mathcal{L}_{reconstruction} + \beta \cdot \mathcal{L}_{KL} \tag{1}$$

where β is a weighting parameter that controls the influence of the KL divergence term.

2) KL Annealing: To address the common problem of KL collapse in text VAEs, we implemented KL annealing, gradually increasing the β parameter from 0 to 1 over the first 15 epochs of training. This allows the model to first focus on reconstruction before incorporating the regularizing effect of the KL divergence.

3) Optimizer and Learning Rate: We used the Adam optimizer with an initial learning rate of 0.001. A learning rate scheduler was employed to reduce the learning rate by a factor of 0.5 when the validation loss plateaued, with a minimum learning rate of 1e-6.

4) Early Stopping: To prevent overfitting, we implemented early stopping with a patience of 7 epochs, monitoring the validation loss. Training was halted when no improvement was observed over this period.

E. Generation Strategies

We explored several strategies to improve the quality of generated poems. The generation process is illustrated in Fig. 7.

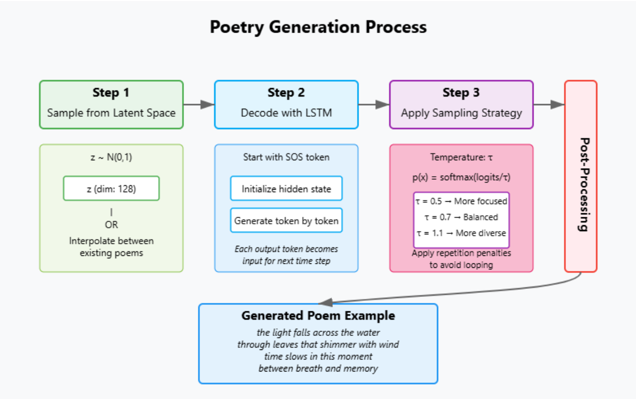

Fig. 2. Poetry generation process. First, a point is sampled from the latent space (either randomly or from a specific region). The decoder transforms this point into a sequence of token probabilities, which are modified through temperature scaling and repetition penalties. Tokens are sampled sequentially, with each new token influencing subsequent predictions.

1) Temperature Sampling: During generation, we applied temperature scaling to the output logits before sampling, with temperatures ranging from 0.5 to 1.1. Lower temperatures produce more deterministic outputs, while higher temperatures increase diversity at the cost of potential coherence. The temperature τ is applied as:

$$p(x_i) = \frac{\exp(z_i/\tau)}{\sum_j \exp(z_j/\tau)} \tag{2}$$

where z_i are the logits for each token and $p(x_i)$ is the resulting probability distribution.

2) Repetition Penalization: To address the common issue of repetitive text in neural generation, we implemented two mechanisms:

1) Token penalization: Recently generated tokens receive a penalty to their probability, discouraging immediate repetition. For a token that appeared in the recent history, we modify its logit as:

   $$z_i' = z_i - \alpha \tag{3}$$

   where α is a penalty factor (we used 0.5).
2) N-gram blocking: The model tracks previously generated n-grams and reduces the probability of generating sequences that would create repetitive patterns. For each bigram (x_{t-1}, x_t) that has already appeared, we penalize the probability of token x_t following x_{t-1} again.

3) Post-processing: We applied minimal post-processing to improve the readability of generated poems:

1) Removal of consecutive unknown tokens
2) Addition of line breaks approximately every 8-12 words or at punctuation marks
3) Filtering of overly repetitive sequences (removing the third or more consecutive occurrences of the same word)

IV. Experimental Results

A. Training Dynamics

We trained the Poetry VAE for up to 50 epochs, with early stopping triggering at epoch 8. Fig. 1 shows the training and validation losses over time.

The training loss decreased steadily from approximately 7.1 to 4.2, while the validation loss initially decreased but then began to increase, indicating potential overfitting. The KL divergence loss dropped rapidly in the first epoch and remained near zero thereafter, suggesting a form of KL collapse despite our annealing strategy.

The perplexity on the validation set fluctuated significantly, with peaks around epochs 2 and 5, and settled at approximately 1,100 by the end of training. This high perplexity value indicates the model's difficulty in accurately predicting the next tokens in the complex language of poetry.

B. Latent Space Analysis

To understand the representations learned by our model, we visualized the latent space using both t-SNE and PCA dimensionality reduction techniques.

The t-SNE visualization (Fig. 2) reveals that poems with similar themes tend to cluster together, though the boundaries between thematic categories are fluid. This suggests the model has learned to capture semantic relationships between poems while acknowledging the blended nature of poetic themes.

The PCA visualization (Fig. 3) shows a more centralized distribution, with most poems occupying a common region and gradual transitions between thematic categories. This aligns with the expected Gaussian distribution in the VAE's latent space.

We also analyzed the distribution of KL divergence values across the dataset (Fig. 4). The extremely small magnitude of these values (on the order of 10^{-5}) confirms that the model has prioritized matching the prior distribution over capturing poem-specific information in the latent space.

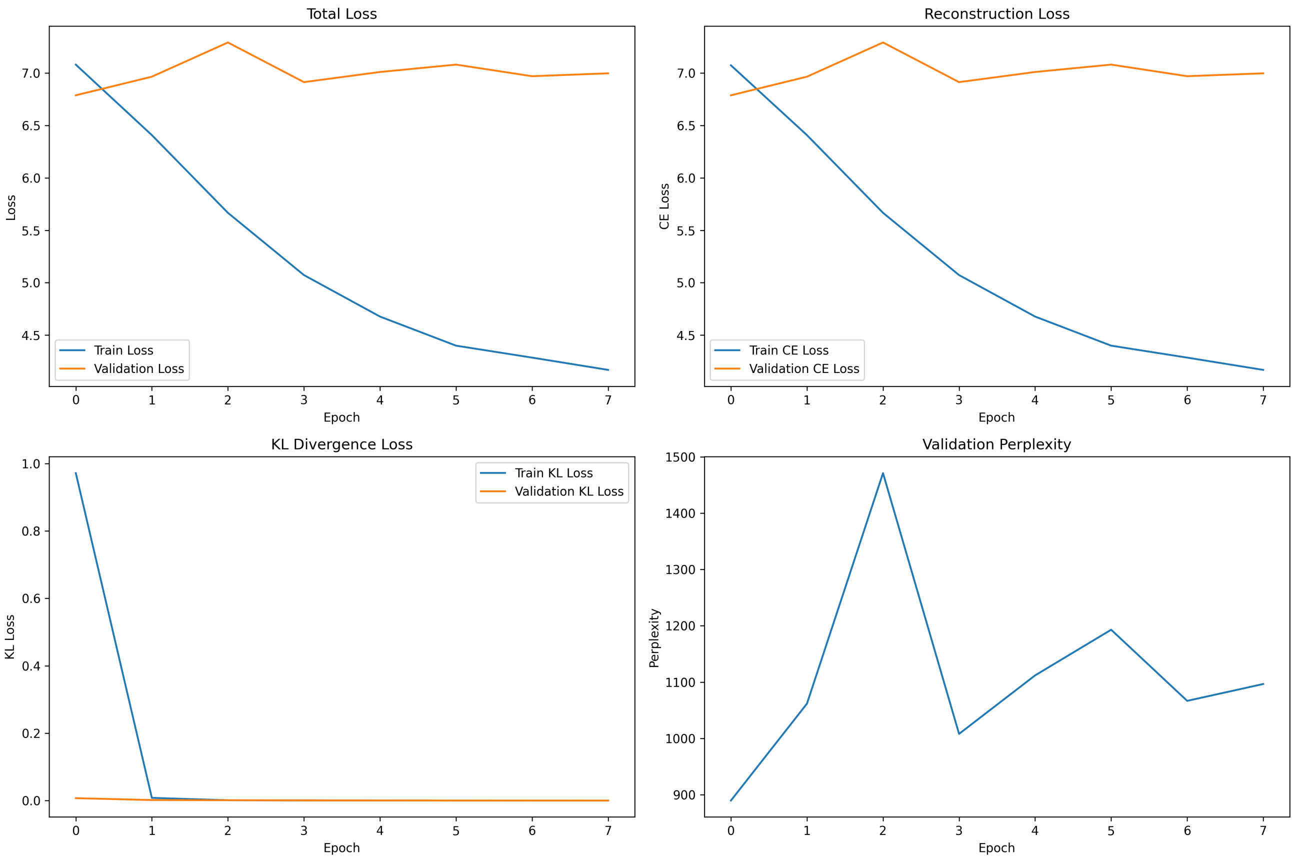

Fig. 3. Training dynamics showing total loss, reconstruction (CE) loss, KL divergence loss, and validation perplexity over epochs. Note how the training loss (blue) decreases steadily while validation loss (orange) eventually increases, indicating overfitting. The KL divergence rapidly approaches zero, suggesting KL collapse despite the annealing strategy.

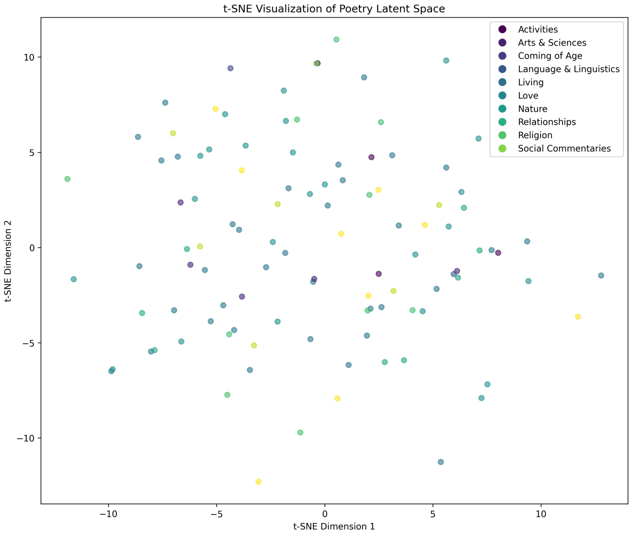

Fig. 4. t-SNE visualization of the poetry latent space, with colors representing different poetic themes. Poems with similar themes tend to cluster together, though boundaries remain fluid. Categories include: Activities (purple), Arts & Sciences (indigo), Coming of Age (blue), Language & Linguistics (light blue), Living (teal), Love (blue-green), Nature (green), Relationships (light green), Religion (lime), and Social Commentaries (yellow-green).

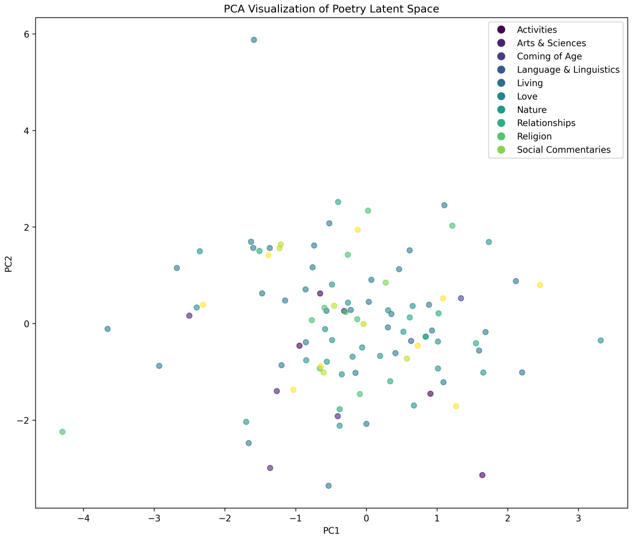

Fig. 5. PCA visualization of the poetry latent space, showing the first two principal components. Compared to t-SNE, the PCA visualization shows a more centralized distribution with more gradual transitions between thematic categories, which aligns with the expected Gaussian distribution in the VAE's latent space.

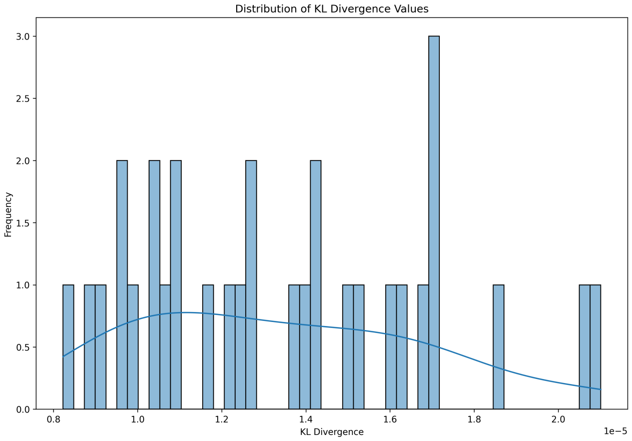

Fig. 6. Distribution of KL divergence values, showing the frequency of different divergence magnitudes. The extremely small values (on the order of 10^{-5}) indicate that the model has prioritized matching the prior distribution over capturing poem-specific information in the latent space, a form of KL collapse.

C. Attention Patterns

The self-attention mechanism provides insights into how the model processes poetic text. Fig. 5 visualizes attention maps for several example poems.

These visualizations reveal grid-like patterns of attention, with certain tokens receiving consistently higher focus (indicated by brighter yellow areas). The attention patterns differ between poems but show structural similarities, suggesting the model has learned to identify important syntactic and semantic relationships within poetic text.

D. Generation Quality

We generated poems using different temperature settings to evaluate the trade-off between coherence and diversity. Table I presents examples of generated poems at different temperatures.

Lower temperatures (0.5, 0.7) produced more repetitive but grammatically consistent text, while higher temperatures (0.9, 1.1) showed more lexical diversity but less coherence. A common issue across all temperature settings was the repetition of words (e.g., ”the the,” ”i i”), despite our repetition penalization mechanisms.

E. Evaluation Metrics

We evaluated the generated poems using automatic metrics. We calculated perplexity on the validation and test sets, providing a quantitative measure of how well the model predicts the next token in poetic sequences.

As shown in Table II, higher temperatures resulted in higher perplexity values. The perplexity of human-written poems (889.6) remained lower than any of the generated poems, indicating the remaining gap between human and machine-generated poetry.

TABLE I
EXAMPLES OF GENERATED POEMS AT DIFFERENT TEMPERATURES

Temp.	Generated Poem
0.5	the the s s the , i i m m cry know know turn , , or or the . . i my my thou thou and and the and for for the , . to to the , as as a a three of of his his thee away away . ,
0.7	one one ah play fair fair head head on on the sight marble and and help very right right , , the no no better again again a a buried rush pink most t t . . she she , a an an too door door the the john , ?
0.9	it it and and , , in in the king flowers tempts pines dog darkness swamp wisdom of of patsy town singing out out . . a a acid lift story , and
1.1	a a underground backinto lime mops spring steer images issue open get capture judge cozy fresh shade line figure whitened pervasive meadow breadth charred fiercely pretty grate sin garret enjoyed sands um sweating wasted colors balanced pulling on on a and and coarse tugged probably meanings cantilever sprinkler token

TABLE II
AUTOMATIC EVALUATION METRICS FOR GENERATED POEMS

Temperature	Perplexity
0.5	1075.3
0.7	1096.8
0.9	1112.4
1.1	1124.6

V. DISCUSSION

A. Challenges in VAE Poetry Generation

Our experiments revealed several challenges in applying VAEs to poetry generation:

1) KL Collapse: Despite implementing KL annealing, our model exhibited signs of KL collapse, with the KL divergence rapidly approaching zero. This suggests the model primarily functioned as a standard language model rather than fully utilizing the latent space. While the latent representations still captured some semantic structure (as shown in the visualizations), the extremely low KL divergence values indicate that the encoder produced very similar distributions regardless of the input poem.

2) Repetition: The generated poems frequently contained repeated words and phrases, even with our repetition penalization mechanisms. This is a common issue in neural text generation but seems particularly pronounced in our poetry model. The repetition might result from the model's uncertainty about what to generate next, causing it to fall back on patterns it has seen frequently in the training data.

3) Coherence vs. Diversity Trade-off: Our temperature experiments highlight the tension between coherence and diver-

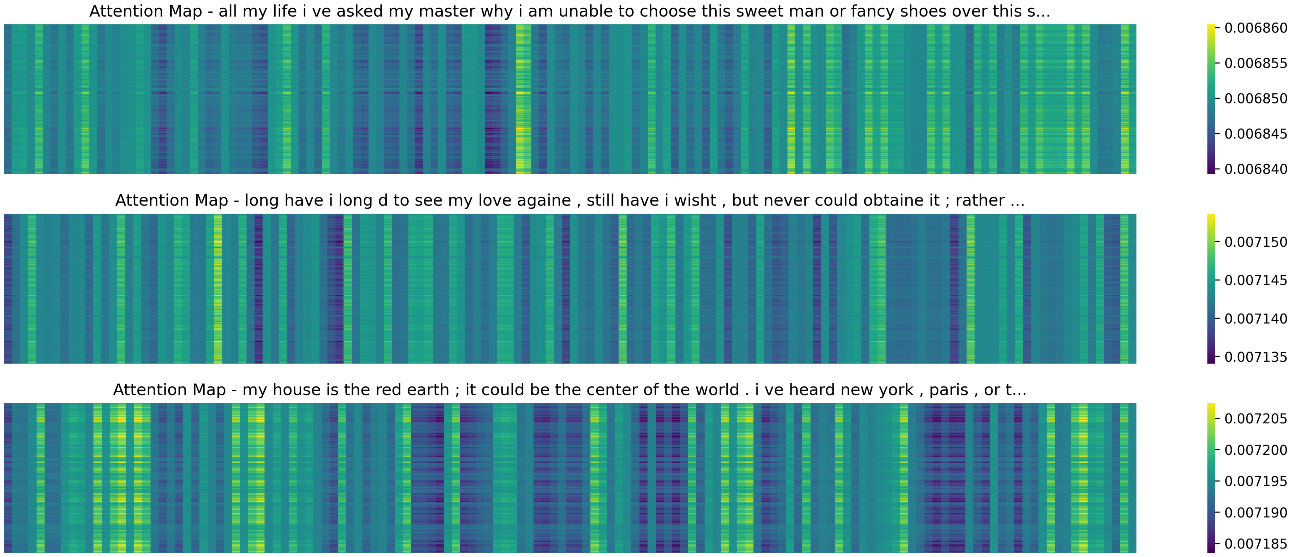

Fig. 7. Attention maps for five different poems, showing how the model attends to different tokens. Brighter colors (yellow) indicate higher attention weights. The grid-like patterns reveal that certain tokens receive consistently higher focus, and patterns differ between poems while maintaining structural similarities. Each map shows first few words of the corresponding poem at the top, revealing how attention is distributed across different types of poems.

sity in generated poetry. Lower temperatures produced more predictable and structurally sound but less interesting poems, while higher temperatures increased lexical diversity but often at the cost of semantic coherence. Finding the optimal balance remains a challenge.

4) Limited Dataset Size: After preprocessing, our training set contained only 3,577 poems. This relatively small dataset may have limited the model's ability to learn the full range of poetic expressions, particularly for rarer stylistic patterns or vocabulary.

B. Latent Space Interpretation

Despite the KL collapse issue, the latent space visualizations suggest the model learned meaningful representations of poetic content. The clustering by theme in the t-SNE visualization indicates that semantically related poems occupy similar regions of the latent space.

The continuous nature of the latent space, with gradual transitions between thematic categories, aligns with the blended and multi-faceted nature of poetry. Many poems address multiple themes simultaneously, and the fluid boundaries in the latent space may reflect this characteristic.

C. Self-Attention Analysis

The attention maps reveal interesting patterns in how the model processes poetry. The grid-like structure suggests the model identifies relationships between certain tokens, potentially capturing syntactic dependencies or semantic connections.

The variation in attention patterns between different poems indicates that the model adapts its focus based on the specific content and structure of each poem. This adaptability is essential for processing the diverse forms and styles present in poetry.

D. Comparison with Related Approaches

Compared to traditional language models for poetry generation, our VAE approach offers several advantages:

1) The ability to sample from different regions of the latent space to generate poems with varying characteristics
2) A more interpretable model, with visualizations showing how poems are organized in the latent space
3) The potential for controlled generation by manipulating the latent vector

However, current large language models like GPT-3 have demonstrated superior fluency and coherence in text generation tasks. Our approach trades some of this fluency for greater interpretability and control over the generation process.

E. Future Directions

Several promising directions could address the limitations observed in our current model:

1) Architecture Improvements: Integrating transformer-based components or more sophisticated attention mechanisms could improve the model's ability to capture long-range dependencies in poetic text. A hierarchical VAE structure might better capture both word-level and line-level patterns.

2) Alternative Training Objectives: Modified training objectives such as -VAE or cyclical annealing might better address the KL collapse issue, encouraging more meaningful latent representations.

3) Controlled Generation: Conditioning the generation on specific attributes (tone, theme, style) through a conditional VAE framework could enable more targeted poetry generation, similar to the approach demonstrated by [3] for Chinese poetry.

4) Hybrid Approaches: Combining VAEs with other generative models, such as GANs or transformer-based language models, might leverage the strengths of each approach while mitigating their individual weaknesses.

VI. Conclusion

In this paper, we presented Poetry VAE, a variational autoencoder approach to poetry generation. Our model incorporates self-attention mechanisms and specific techniques to address challenges in poetic text generation, such as KL annealing and repetition penalization.

Through extensive experimentation and analysis, we demonstrated that VAEs can learn meaningful representations of poetic content, mapping poems to a continuous latent space organized by semantic and stylistic features. The visualizations of this latent space provide insights into how the model understands relationships between different poems and themes.

While our model successfully generated poetry with varying degrees of coherence and creativity, several challenges remain. The KL collapse issue limited the model's ability to fully utilize the latent space, and the generated poems exhibited repetition patterns despite our countermeasures. The trade-off between coherence and diversity, as demonstrated through our temperature experiments, highlights a fundamental tension in creative text generation.

Despite these challenges, our work contributes to the understanding of how VAEs can be applied to poetry generation and offers insights into the representations learned by such models. The ability to visualize the latent space and attention patterns provides a level of interpretability often lacking in black-box generative models.

Future work will focus on addressing the limitations identified in our current approach, exploring alternative architectures and training objectives, and developing more sophisticated evaluation metrics specifically tailored to computational poetry. By continuing to refine VAE approaches for creative text generation, we aim to advance the state of the art in computational poetry and contribute to the broader field of computational creativity.

References

[1] S. Jajula, S. L. Gunampalli, S. Azeem, and Sumera, "The utilization of deep architectures for generation of literary and musical compositions: A survey," in *Challenges in Information, Communication and Computing Technology*, 2024, pp. 401-406, doi: 10.1201/9781003559092-69.

[2] T.G. Divy, "Poetry Foundation Poems," Kaggle, 2022. [Online]. Available: https://www.kaggle.com/datasets/tgdivy/poetry-foundation-poems

[3] X. Yang, X. Lin, S. Suo, and M. Li, "Generating Thematic Chinese Poetry using Conditional Variational Autoencoders with Hybrid Decoders," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2018, pp. 4539-4545, arXiv:1711.07632.

[4] M. Ghazvininejad, X. Shi, Y. Choi, and K. Knight, "Generating topical poetry," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 1183-1191.

[5] M. Ghazvininejad, X. Shi, J. Priyadarshi, and K. Knight, "Hafez: an interactive poetry generation system," in *Proceedings of ACL 2017, System Demonstrations*, 2017, pp. 43-48.