An Enhanced Probabilistic Approach for Singfake Generation

Arth J. Shah*, Aniket Pandey*, Satyam R. Tiwari†, and Hemant A. Patil*
* Speech Research Lab, Dhirubhai Ambani University (DAU)
E-mail: {202101154, 202411001, hemant_patil}@dau.ac.in
† Sarvajanik College of Engineering and Technology (SCET), Surat
E-mail: satyamtiwari.co22d2@scet.ac.in

***Abstract*—This work proposes an enhanced probabilistic framework for singing voice deepfake generation using Variational Autoencoders (VAEs) augmented with Kolmogorov–Arnold Network (KAN) variants. The goal is to generate high-fidelity and controllable singing voices by disentangling singer identity and acoustic content in the *latent* space. We explore multiple KAN-based bottleneck architectures—B-Spline Radial Basis Function (BSRBF) KAN, Chebyshev-KAN, and WaveletKAN(WavKAN)—with and without Gaussian Error Linear Unit (GELU) activation, to improve the expressiveness of the latent representation. Experiments conducted on Mel spectrogram representations of singing audio reveals that WavKAN-GELU achieves the best performance, with relatively lowest Mel Cepstral Distortion (15.25), Mean Spectral Distortion (0.065), and high correlation (0.9436) and cosine similarity (0.993). Compared to the baseline VAE (MCD = 21.75, MSD = 0.108, correlation = 0.846), WavKAN-GELU significantly enhances generation quality and alignment with real data. The BSRBF-KAN-GELU variant also shows competitive results with an MCD of 17.91, and correlation of 0.9399. These results demonstrate that wavelet and polynomial-based nonlinearities in the bottleneck improve both reconstruction and generative capabilities. Our model supports interpretable latent control and opens avenues for robust singing deepfake synthesis and detection.**

I. Introduction

The recent advancements in generative modeling have enabled the synthesis of highly realistic audio, including singing voice generation that mimics the identity, style, and expressive nuances of human singers. Among various techniques, deep generative models have emerged as powerful tools for modeling complex data distributions, particularly in high-dimensional domains, such as music and speech. However, producing high-quality and controllable singing voice deepfakes requires not only generating realistic acoustic features but also learning disentangled and interpretable representations that reflect underlying factors, such as singers identity, pitch (F_0) contour, and timbre. **V**ariational **A**uto**E**ncoders (VAEs) [1], with their principled probabilistic framework, offer a promising approach to this challenge. VAEs address a fundamental limitation of traditional autoencoders by introducing a stochastic and continuous latent space that enables both data reconstruction and generation. While conventional autoencoders compress inputs into deterministic latent vectors [2], VAEs learn a distribution over the latent space, allowing smooth interpolation and sampling. This generative capability is particularly valuable in the context of deepfake singing synthesis, where the ability to manipulate latent variables can facilitate tasks, such as identity transfer, prosody control, and expressive singing style imitation. The training objective of VAEs, grounded in variational inference, optimizes the **E**vidence **L**ower **BO**und (ELBO), balancing reconstruction accuracy and regularization through **K**ullback-**L**eibler (KL) divergence [3]. This ensures that the learned latent space remains structured and aligned with a prior distribution, typically a multivariate Gaussian.

In contrast to alternative models, such as Generative Adversarial Networks (GANs) [4], which often produce sharper (or fine) outputs but lack interpretability and training stability, VAEs provide a more tractable and explainable generative process. Their smooth latent space supports robust sampling and fine-grained control over generated outputs, making them especially suitable for audio domains, where subtle variations in articulation and expression are critical. Furthermore, the probabilistic encoding-decoding mechanism of VAEs allows for flexible integration with downstream components, such as vocoders or neural synthesizers, enabling end-to-end systems for singing voice cloning and transformation. In this work, we leverage the capabilities of VAEs for the task of singing deepfake generation [5]. We propose a framework that learns to disentangle singer-specific and content-related features from sung audio, enabling the synthesis of novel singing voices that preserve the melody and rhythm of the source while adopting the timbre and stylistic characteristics of a target identity. By employing a variational latent space, our system allows for intuitive manipulation and sampling, which opens avenues for data augmentation, identity morphing, and detection of synthetic singing. Through rigorous evaluation on a diverse language based audio dataset of singing recordings, we demonstrate that our VAE-based approach achieves high fidelity in both reconstruction and generation, while maintaining interpretability and controllability essential for analysis and its' applications. All the codes are publicly available on [1]

A. Related Works

Several notable efforts have shaped the trajectory of research in recent years. In the synthesis domain, VISinger introduced a VAE-inspired end-to-end architecture that employs varia-

[1] https://github.com/ARTHARKING55/SingVAE

979-8-3315-7206-8/25/$31.00

tional inference and adversarial learning to generate high-fidelity singing voice waveforms directly from the lyrics and musical scores, achieving significantly improved naturalness and expressiveness compared to pipeline approaches [6]. Concurrently, diffusion-based models such as DiffSinger have demonstrated the effectiveness of probabilistic generative techniques for singing voice synthesis; by optimizing a variational bound through a shallow diffusion mechanism, DiffSinger produces realistic audio while maintaining training stability [7]. Shifting focus toward security concerns, the SingFake benchmark was introduced as the first large-scale dataset for singing voice deepfake detection, revealing that state-of-the-art speech-based countermeasure systems struggled to generalize to singing audio and highlighting challenges in detecting deepfakes amidst complex musical contexts [5]. More recently, the CtrSVDD dataset has extended this work by offering controlled conditions, diverse synthesis methods, and custom features tailored for singing deepfake detection, striving to improve generalizability across varying generation techniques [8]. This study offers the following novelty:

- VAE for singfake generation.
- KAN + Gated weights for VAE.
- End-to-End singfake generation.

II. Proposed Methodology

A. Variational Autoencoder

Variational autoencoders (VAEs) are probabilistic neural models, where an encoder maps input data (x) to parameters-mean [$\mu(x)$] and variance [$\sigma^2(x)$] of a latent distribution (z), typically a Gaussian, and a decoder reconstructs the input($\hat{x}$) by sampling from this distribution. A KL divergence term in the loss aligns the latent space with a prior, enabling generative capabilities.

Unlike standard autoencoders, which focus on deterministic reconstruction for tasks, such as compression, VAEs offer *data generation* by sampling from the prior, supporting applications, such as image synthesis. The reparameterization trick ensures differentiability, facilitating end-to-end optimization. This probabilistic framework provides VAEs with a structured latent space, enhancing interpretability and generative performance over autoencoders.

B. Preprocessing Pipeline

Raw audio inputs in various formats are first resampled to a uniform sampling rate of 22.05 kHz and organized for consistent downstream processing. Then, we obtain Mel spectrograms from the input audio files using `Librosa` library in Python. For visualization and storage, these spectrograms are often converted to RGB images using a colormap (e.g., `viridis`, `magma`), resulting in a tensor of shape $128 \times 128 \times 3$. However, these three channels do not represent distinct acoustic features; they are color-mapped encodings of a single intensity value per time-frequency bin.

The spectrograms are then preprocessed by resizing or cropping them to a fixed dimension of 128×128. Subsequently, normalization is applied—either through min-max scaling to the range $[0, 1]$ or standardization to zero mean and unit variance—to generate a single-channel normalized tensor of shape $128 \times 128 \times 1$. This normalized tensor serves as the input to the encoder of the VAE.

C. Encoder

The encoder of a VAE maps the input $x \in \mathbb{R}^d$ to parameters of a probability distribution function over the latent space $\mathbf{z} \in \mathbb{R}^k$. Typically, this involves predicting the mean $\boldsymbol{\mu}$, and the logarithm of the variance, $\log \boldsymbol{\sigma}^2$ of a multivariate Gaussian distribution.

In our architecture, we explicitly decouple deterministic representation learning and probabilistic modeling by introducing a separate bottleneck layer after the encoder. The encoder is responsible for extracting high-dimensional features, while the bottleneck independently infers the variational parameters $\boldsymbol{\mu}$ and $\log \boldsymbol{\sigma}^2$. This modular design increases the flexibility and interpretability of the system.

The encoder receives a single-channel Mel spectrogram input of shape $[B, 1, 128, 128]$, where B denotes the batch size. It consists of four convolutional blocks, each comprising a 2D convolution [9], ReLU activation, batch normalization, and dropout for regularization. The stride of each block increases spatial compression progressively, reducing the dimensionality from 128×128 to 16×16 over four stages while increasing the number of channels. The final output tensor of shape $[B, 256, 16, 16]$ is flattened to a vector of size $[B, 65536]$.

To enhance the nonlinearity and expressive capacity of the encoder, we integrate **GELU Layers** after selected convolutional blocks. These layers employ the Gaussian Error Linear Unit (GELU) [10], which improves model generalization and learning dynamics compared to traditional activations, such as ReLU. The GELU activation is defined as:

$$\text{GELU}(x) = x \cdot \Phi(x), \tag{1}$$

where $\Phi(x)$ is the cumulative distribution function of the standard normal distribution. Unlike ReLU, which abruptly zeroes out negative inputs, GELU allows small negative activations to pass through, enabling smoother gradient flow and better handling of uncertainty in the feature space—properties that are particularly beneficial in probabilistic models, such as VAEs. The inclusion of GELU gated layers enables the encoder to learn richer and more stable feature representations, which improves both the quality of the latent variables and the fidelity of the reconstructed audio after decoding. Algorithm 1 displays the encoder architecture employed for this particular study.

D. Bottleneck

The bottleneck layer serves as the probabilistic component of the variational autoencoder. It takes as input the flattened deterministic feature vector of shape $[B, 65536]$ produced by the encoder. This vector is passed through two parallel fully-connected layers to produce the parameters of a multivariate Gaussian distribution in the latent space $\mathbb{R}^k$, where k is 128. Specifically, the bottleneck computes the mean vector

Algorithm 1 Vanilla VAE Encoder with Optional GELU Gated Layers

$x \leftarrow$ input tensor $[B, 1, 128, 128]$
for $i = 1$ to N **do**
  $x \leftarrow \text{Conv2D}(x, \text{filters}[i], \text{kernel}[i], \text{stride}[i], \text{padding})$
  $x \leftarrow \text{ReLU}(x)$
  $x \leftarrow \text{BatchNorm}(x)$
  $x \leftarrow \text{Dropout}(x)$
  {Optional GELU Gated Layer}
    $x \leftarrow \text{Permute}(x, [B, H, W, C])$
    $x \leftarrow \text{Reshape}(x, [BHW, C])$
    $x \leftarrow \text{GELUGatedLayer}(x)$
    $x \leftarrow \text{ReshapeBack}(x, [B, H, W, C])$
    $x \leftarrow \text{PermuteBack}(x, [B, C, H, W])$
end for
$x \leftarrow \text{Flatten}(x)$
$\mu \leftarrow \text{Linear}(x \rightarrow \text{latent\_dim})$
$\log \text{var} \leftarrow \text{Linear}(x \rightarrow \text{latent\_dim})$
return $\mu, \log \text{var}$

$\boldsymbol{\mu} \in \mathbb{R}^{B\times 128}$ and the log-variance vector $\log \boldsymbol{\sigma}^2 \in \mathbb{R}^{B\times 128}$ as follows:

$$\boldsymbol{\mu} = W_\mu h + b_\mu, \tag{2}$$

$$\log \boldsymbol{\sigma}^2 = W_{\log \sigma} h + b_{\log \sigma}, \tag{3}$$

where W_μ and $W_{\log \sigma}$ are learnable weights to obtain the mean and log-variance vectors, respectively, b_μ and $b_{\log \sigma}$ are learnable bias vectors for mean and log-variance, respectively, and $h \in \mathbb{R}^{B\times 65536}$ denotes the flattened output of the encoder for a batch of size, B.

1) KAN-Based Bottleneck Layer: To enhance the expressiveness of the bottleneck layer, we replace the traditional linear transformation with three different variants of KAN: *BSRBF-KAN Layer*[11], *ChebyshevKAN Layer*[12], and *WavKAN Layer*[13]. These modules are designed to capture complex, nonlinear interactions that standard fully-connected layers cannot model effectively.

The conventional bottleneck in a VAE uses a linear layer to map a high-dimensional encoder output to the mean $\boldsymbol{\mu}$ and log-variance $\log \boldsymbol{\sigma}^2$ of the latent Gaussian distribution. However, this assumes an affine mapping, which may limit the model's ability to capture intricate patterns in audio spectrogram data. To address this, we incorporate learnable nonlinear basis expansions that better approximate the true posterior distribution.

BSRBF-KAN Layer introduces a smooth interpolation of the input space using B-spline and radial basis functions (RBFs)[14], enabling the model to focus on local and smooth variations in the encoder output. The basis functions are differentiable and compactly supported, promoting sparse representations.

The transformation output of BSRBF-KAN layer is computed as follows:

$$\mathbf{y}_{\text{BSRBF}} = W_1 h + W_2(RBF(h) + Spline(h)), \tag{4}$$

where h is the flattened encoder output, W_1 and W_2 are learnable weight matrices, $RBF(\cdot)$ denotes a Gaussian radial basis function, and $Spline(\cdot)$ is the B-Spline transformation capturing additional nonlinearity.

ChebyshevKAN Layer applies Chebyshev polynomial expansions to the input, which are known to minimize the maximum error of polynomial interpolation. These orthogonal polynomials provide a global approximation that is robust to noise and effective in capturing global trends. The transformation output of ChebyshevKAN layer is computed as follows:

$$\mathbf{y}_{\text{chebyshev}} = \alpha \cdot W_1 h + \beta \cdot W_2 \cdot T(h), \tag{5}$$

where h is the flattened encoder output, W_1 and W_2 are learnable weight matrices, $T(\cdot)$ represents Chebyshev polynomial basis functions, α and β serve as the scalar hyperparameters.

In our experiments, we set $\alpha = \beta = \mathbf{1}$, as this choice maintains an equal balance between the base and nonlinear components, enabling the model to leverage both linear and polynomial expressivity without introducing scaling bias.

WavKAN Layer incorporates wavelet-based transformations to introduce non-linearity. In this experiment, we have used **D**erivative **o**f **G**aussian (DoG) as mother wavelet. This allows the model to perform multi-scale analysis of input features, which is especially beneficial for audio data with hierarchical time-frequency structures.

The equation for m^{th} order DoG wavelet is as follows:

$$\Psi_{a,b}(t) = \frac{d^m}{dt^m}\left(e^{-\frac{(t-b)^2}{2a^2}}\right), \tag{6}$$

where $\Psi_{a,b}(t)$ is the DoG wavelet basis function at scale a and translation b, $a > 0$ controls the scale (dilation) of the wavelet, and $b \in \mathbb{R}$ represents the shift (translation) in time. For this experiment. we have taken the 1^{st} order DoG wavelet (m = 1). The transformation output of WavKAN layer is computed as follows:

$$\mathbf{y}_{\text{wavelet}} = W_1 h + W_2 \cdot \Psi(h), \tag{7}$$

where h is the flattened encoder output, W_1 and W_2 are learnable weight matrices, and $\Psi(\cdot)$ represents DoG wavelet basis function.

By enriching the latent transformation with domain-specific basis expansions, each KAN variant improves the model's capacity to approximate more complex posterior distributions in the latent space. These modifications result in better generalization and reconstruction performance compared to standard linear VAEs.

To enable stochastic sampling during training while preserving differentiability, the *reparameterization trick* is employed. A sample $\mathbf{z}$ from the posterior distribution is obtained as:

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \tag{8}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ has the same shape as the standard deviation. The noise enables stochastic sampling from the latent space while maintaining the gradient flow during backpropagation. The standard deviation $\boldsymbol{\sigma}$ can be obtained by:

$$\boldsymbol{\sigma} = \exp\left(0.5 \cdot \log \boldsymbol{\sigma}^2\right), \tag{9}$$

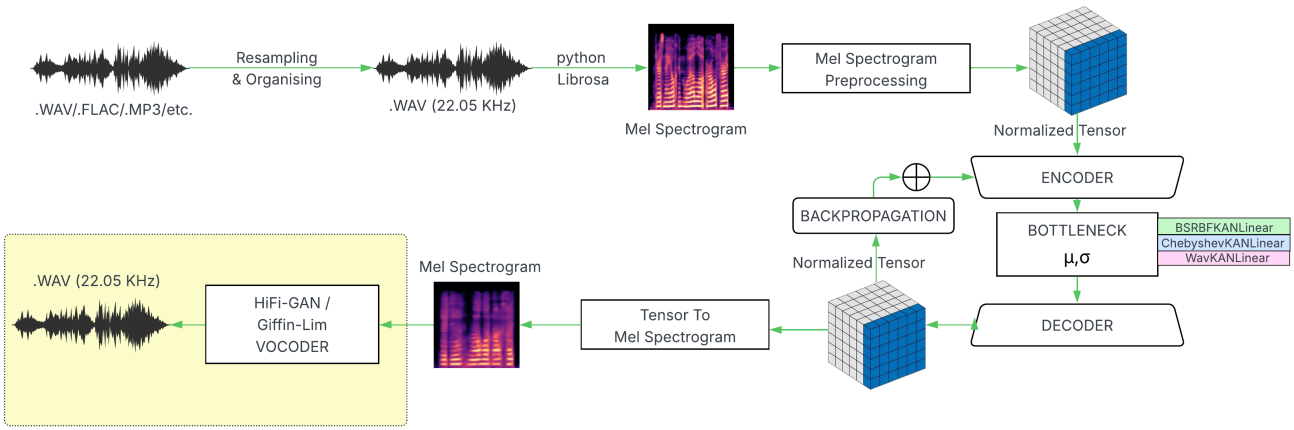

Fig. 1. Pipeline of the proposed architecture (yellow block represents the future plans).

where $\log \boldsymbol{\sigma}^2$ is the log-variance vector computed by the bottleneck layer.

This reparameterized latent vector $\mathbf{z} \in \mathbb{R}^{B\times 128}$ is then forwarded to the decoder. By isolating the probabilistic modeling into the bottleneck layer, the design cleanly separates deterministic representation learning from variational inference. This modularity promotes architectural flexibility, allowing the encoder to be enhanced independently while maintaining the probabilistic semantics of the latent space in the bottleneck.

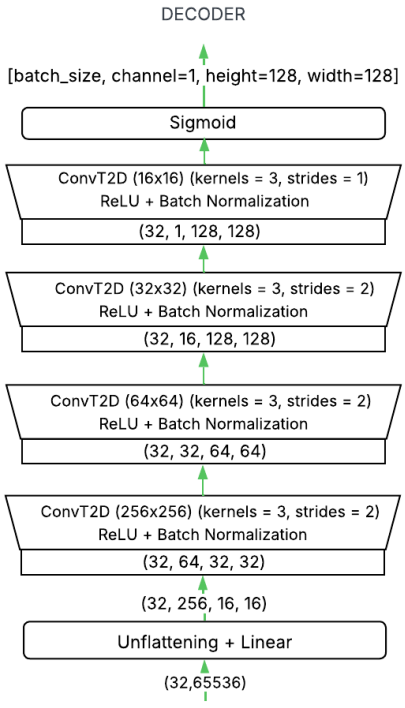

Fig. 2. Decoder Architecture

E. Decoder

The decoder aims to reconstruct the Mel spectrogram from the latent variable $\mathbf{z}$, which has the shape $[B, 128]$, where B denotes the batch size and 128 is the latent dimension. It takes as input the sampled latent representation and learns to invert the hierarchical compression performed by the encoder as displayed in Fig. 2.

The decoder begins with a fully-connected layer that projects the latent vector $\mathbf{z}$ into a high-dimensional feature space of shape $[B, 65536]$. This output is then reshaped (unflattened) to a 4D tensor of shape $[B, 256, 16, 16]$, which mirrors the final feature map size of the encoder. Subsequently, the decoder employs a series of transposed convolutional layers [9] with decreasing filter dimensions to progressively upsample the feature maps back to the original input resolution. Each Conv2DTranspose layer uses a kernel size of 3 and a stride of either 2 (for upsampling) or 1 (for resolution preservation), and is followed by a ReLU activation and batch normalization. A final sigmoid activation function is applied to constrain the output values within the range $[0, 1]$, aligning with the normalized Mel spectrogram scale. The decoder thus reconstructs a tensor of shape $[B, 1, 128, 128]$, which is the final output of our VAE model, designed to closely approximate the input mel spectrogram. The VAE is trained to minimize a combination of reconstruction loss and KL divergence, ensuring both accurate reconstruction and a well-regularized latent space.

F. Dataset Used and Training Parameters

The real data for training of VAE model was taken from SingFake dataset [15] (all training and validation real samples were accounted). The VAE models are trained for 250 epochs using the Adam optimizer with an initial learning rate of $1e{-}3$ and a batch size of 32. The reconstruction loss is computed using Mean Squared Error (MSE), and the KL divergence is scaled with a weight annealing schedule. Additionally, variants of the VAE architecture replace the standard Linear layer in the bottleneck with advanced formulations, such as

TABLE I
RESULTS OBTAINED ON VARIOUS KANS BASED VAES.

Model	Frechet Distance	MCD	MSD	MAE	Correlation	Cosine Similarity	Coherence	Spectral Convergence	MSE	Loss
VAE	7.1037	21.7454	0.1077	0.2347	0.8458	0.8457	0.4002	0.5374	0.1077	193.3154
GELU	**3.5299**	17.3243	0.0743	0.1847	**0.952**	**0.994**	**0.4536**	0.4413	0.0743	**113.8874**
WavKAN	3.534	17.1486	0.0734	0.182	0.9019	0.9019	0.4522	0.4379	0.0734	203.7878
WavKAN-GELU	3.7242	**15.2472**	**0.0652**	**0.162**	**0.9436**	**0.993**	**0.4467**	**0.3913**	**0.0652**	121.6411
Chebyshev-KAN -GELU	6.4112	21.5602	0.1105	0.2418	0.9934	0.9473	0.3898	0.5401	0.1105	230.7312
Chebyshev-KAN	35.215	41.6783	0.4118	0.4669	0.4713	0.4716	0.4044	1.0377	0.4118	462.1069
BSRBF-KAN	5.4259	21.0199	0.1063	0.2335	0.8579	0.8579	0.3935	0.5288	0.3913	138.0969
BSRBF-KAN-GELU	3.7242	17.905	0.0757	0.1892	0.9399	0.9925	0.4467	0.4453	0.0757	122.5216

BSRBFKAN, ChebyshevKAN, and WavKAN layers (with and without GELU gating), to evaluate their efficacy in modeling nonlinearities.

III. EXPERIMENTAL RESULTS

The results provided in Table I compares the performance of various models—including a standard VAE and several KAN variants—on multiple objective metrics relevant to the task of singing deepfake generation and evaluation. Here's an explanation of the results with reference to the descriptions in the provided paper: The baseline VAE model shows reasonable performance but is generally outperformed by most of the KAN-enhanced variants. For example, the VAE has a Frechet Distance of 7.1037, which indicates the generated singing voice distributions are farther from the real distribution compared to the other models. Its MCD of 21.7454 and MSD of 0.1077 are also higher, suggesting less accurate spectral reconstruction. The MAE is 0.2347, and the correlation and cosine similarity values (0.8458) are lower than those of enhanced models, indicating that the VAE reconstructs features with less precision and alignment. Despite these limitations, the VAE offers a coherent latent space suitable for generation, which is its core strength. WavKAN, a KAN-enhanced variant using wavelet-based transformations in the bottleneck layer, delivers the best balance of generative quality and signal fidelity. With a Frechet Distance of 3.534, MCD of 17.1486, and MSD of 0.0734, it surpasses the VAE by a wide margin. These improvements are due to WavKAN's ability to capture fine-grained time-frequency characteristics of the audio through derivative-of-Gaussian wavelets, which help the model perform hierarchical analysis on the input signal [16]. Its correlation and cosine similarity values are around 0.9019, confirming a stronger alignment between generated and ground truth spectrograms. The Chebyshev-KAN-GELU variant applies Chebyshev polynomial expansions along with GELU activations. While it achieves an exceptionally high correlation of 0.9934, indicating almost perfect statistical dependence, its spectral convergence is the worst (1.0377), and MCD/MSD values are also higher than WavKAN or GELU. This indicates that although the model aligns closely in terms of global features, it may overfit or lose fine structural details in the waveform. BSRBF-KAN-KAN and BSRBF-KAN-GELU utilize smooth interpolative basis functions, such as B-splines and RBFs. These models generally perform well, with metrics sitting between the baseline VAE and the top-performing WavKAN variants. The BSRBF-KAN-KAN model, in particular, shows a lower loss (138.0969) than the VAE, indicating better convergence during training.

The GELU-only model (which presumably includes GELU layers without KAN enhancements) also performs competitively, with a Frechet Distance of 3.5299, nearly matching WavKAN. Its cosine similarity is the highest overall (0.994), showing excellent angular alignment in the latent feature space. Its low loss value of 113.8874 reflects good training efficiency and stability. One outlier in the Table I is Chebyshev-KAN, which performs extremely poor across most of the metrics—its Frechet Distance is 35.215, and MCD is 41.6783, suggesting that this model variant fails to capture meaningful representations, likely due to overfitting or numerical instability in the polynomial expansions. Lastly, WavKAN-GELU, a hybrid model combining WavKAN's wavelet expressiveness with GELU's smooth nonlinearity, achieves the best overall results. It boasts the lowest MCD (15.2472) and MSD (0.0652), suggesting the most accurate spectral reconstruction, and a Frechet Distance of 3.7242, competitive with the best-performing models. Its loss value of 121.6411 reflects an efficient training regime, while maintaining high correlation (0.9436) and cosine similarity (0.993), which indicate both fidelity and alignment in generation.

A. Training Loss Analysis

The training loss plot compares the convergence behavior of the baseline VAE model with its KAN-enhanced variants over 250 epochs. Notably, WavKAN-GELU and GELU-only models demonstrate the fastest and most stable convergence, reaching minimal loss values early and maintaining consistent performance, indicating strong learning dynamics and effective reconstruction capability. The baseline VAE, along with BSRBF-KAN and WavKAN, also shows smooth convergence but at a slower rate and with higher final loss values. In contrast, Chebyshev-KAN exhibits the poorest performance, with a slow and inconsistent decline in loss, suggesting instability and overfitting likely caused by the sensitivity of high-order polynomial expansions. The inclusion of GELU activation significantly boosts the performance across all KAN variants, especially in BSRBF-KAN-GELU and WavKAN-GELU, underscoring the benefit of smoother non-linear activation in enhancing gradient flow and generalization. Overall, the plot

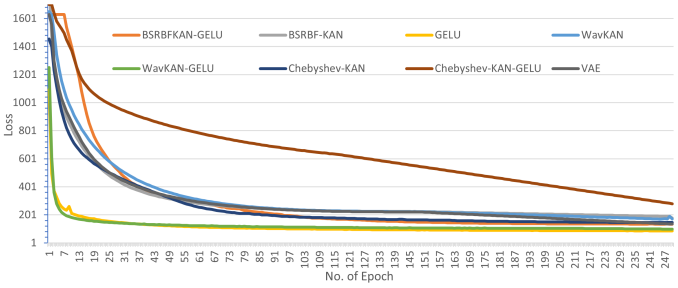

Fig. 3. Loss variation while training of model after each epoch.

in Fig. 3 highlights that incorporating domain-specific basis functions and GELU gating improves the model's ability to minimize reconstruction loss efficiently and robustly.

IV. Summary and Conclusions

This paper proposed a VAE-based framework for singing voice deepfake generation, enhanced with KAN variants—BSRBF-KAN, Chebyshev-KAN, and WavKAN—to improve the modeling of complex latent distributions. Among all variants, the WavKAN-GELU model achieved the best results, with a MCD of 15.25, MSD of 0.0652, and correlation and cosine similarity above 0.94 and 0.99, respectively, significantly outperforming the baseline VAE. While the models demonstrated strong reconstruction and generative quality, limitations remain. The reliance on Mel spectrograms may lead to loss of fine temporal and phase details, and certain KAN variants, such as, Chebyshev-KAN, showed instability and overfitting. Our future work will explore integrating Fourier Analysis Networks (FANs) into the architecture to better capture *harmonic* structures and frequency dynamics. FANs can provide more compact and expressive frequency-domain representations, improving both synthesis fidelity and model generalization. In future, we also plan to generate a pipeline and deploy model in form of API, which can be accessed to generate singfakes via mobile phones also.

References

[1] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in 2^{nd} *International Conference on Learning Representations, (ICLR), Banff, AB, Canada*, Y. Bengio and Y. LeCun, Eds., 2014.

[2] W. Wang, Y. Huang, Y. Wang, and L. Wang, "Generalized autoencoder: A neural network framework for dimensionality reduction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2014, Columbus, OH, USA, pp. 490–497.

[3] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[5] Y. Zang, Y. Zhang, M. Heydari, and Z. Duan, "Singfake: Singing voice deepfake detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, Seoul, Korea, pp. 12 156–12 160.

[6] Y. Zhang, J. Cong, H. Xue, L. Xie, P. Zhu, and M. Bi, "Visinger: Variational inference with adversarial learning for end-to-end singing voice synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, Singapore, pp. 7237–7241.

[7] J. Liu, C. Li, Y. Ren, F. Chen, and Z. Zhao, "Diffsinger: Singing voice synthesis via shallow diffusion mechanism," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, 2022, pp. 11 020–11 028.

[8] Y. Zang, J. Shi, Y. Zhang, *et al.*, "Ctrsvdd: A benchmark dataset and baseline analysis for controlled singing voice deepfake detection," in *INTERSPEECH, Kos, Greece, 2024*, pp. 774–781.

[9] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint arXiv:1603.07285*, 2016, {Last Accessed: $2^{nd} July, 2025$}.

[10] M. Lee, "Mathematical analysis and performance evaluation of the gelu activation function in deep learning," *Journal of Mathematics*, vol. 2023, no. 1, p. 4 229 924, 2023.

[11] H.-T. Ta, "Bsrbf-kan: A combination of b-splines and radial basis functions in kolmogorov-arnold networks," in *International Symposium on Information and Communication Technology (ISCIT), 2024, Bangkok, Thailand*, pp. 3–15.

[12] S. SS, K. AR, A. KP, *et al.*, "Chebyshev polynomial-based Kolmogorov-Arnold Networks: An efficient architecture for nonlinear function approximation," *arXiv preprint arXiv:2405.07200*, 2024, {Last Accessed: 2^{nd} June, 2025}.

[13] Z. Bozorgasl and H. Chen, "WAV-KAN: Wavelet kolmogorov-arnold networks," *arXiv preprint arXiv:2405.12832*, 2024, {Last Accessed: 2^{nd} June, 2025}.

[14] C. S. K. Dash, A. K. Behera, S. Dehuri, and S.-B. Cho, "Radial basis function neural networks: A topical state-of-the-art survey," *Open Computer Science*, vol. 6, no. 1, pp. 33–63, 2016.

[15] Y. Zang, J. Shi, Y. Zhang, *et al.*, "CtrSVDD: A benchmark dataset and baseline analysis for controlled singing voice deepfake detection," in *INTERSPEECH 2024, Kos, Greece*, I. Lapidot and S. Gannot, Eds.

[16] A. J. Shah and H. A. Patil, "Significance of lower frequency regions for audio deepfake detection," in *Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Macau, China*, 2024, pp. 1–6.