EavaNet: Enhancing Emotional Facial Expressions in 3D Avatars through Speech-Driven Animation

Seyun Um*, Yongju Lee*, WooSeok Ko*, Yuan Zhou†, Sangyoun Lee*, Hong-Goo Kang*
* Department of Electrical and Electronic Engineering , Yonsei University, Seoul, Korea
E-mail: [syum,wsko]@dsp.yonsei.ac.kr, [paulyongju,syleee,hgkang]@yonsei.ac.kr
† 01.AI, Beijing, China
E-mail: zhouyuan@01.ai

***Abstract*—Speech-driven 3D facial animation models are essential for creating human-like avatars that synchronize lip movements realistically with speech. Despite these advancements, it is still difficult to effectively convey a wide range of emotional facial expressions that align with the voice. This issue arises due to the lack of clearly labeled datasets for individual emotions as well as insufficient inputs to adequately describe these emotions. To overcome this challenge, we propose a re-categorization process that reduces the data into four emotional groups: angry, sadness, happy, and neutral. We use the re-categorized datasets to estimate style embeddings, which serve to distinctly express emotions and control their intensity. Additionally, we tackle the challenge of slow inference speed in autoregressive models by introducing EavaNet, a non-autoregressive model utilizing gated activation units (GAUs) and bidirectional long short-term memory (BLSTM) modules for efficient prediction of 3D face mesh vertices. Our proposed model outperforms previous state-of-the-art models in terms of emotional expressiveness and lip synchronization accuracy in both subjective and objective evaluations.** [1]

I. Introduction

Speech-driven 3D facial animation systems are now widely used in personal assistants, computer games, film production, and virtual communication systems. Recent developments in artificial intelligence and 3D face modeling technologies have made it easier to create lifelike avatars capable of accurately synchronizing voice and lip movements[1]. However, most 3D facial animation models have primarily focused on improving lip synchronization rather than the full range of facial movements associated with emotions conveyed by the voice. As a result, these systems can generate unnatural facial expressions that do not match the emotions expressed in the voice, which limits their usefulness.

In this work, we identify two primary reasons for this issue: low-quality training data labels and the lack of style embeddings with facial expression information. Previous state-of-the-art models [2], [3] have used 3D subject IDs as supplementary input to produce speaker-dependent facial animations. Although these models can create 3D avatars with high lip synchronization accuracy, they often generate monotonous facial expressions and fail to properly convey the emotional nuances present in the input audio.

In our experiments, we utilize the BIWI dataset [4], which includes emotional speech paired with corresponding expressive facial movements, aligning with our research objectives. However, this dataset has limitations; the emotion labels lack consistency across different subjects, and its size is insufficient for effectively training. To address these challenges, we reclassify the BIWI dataset into four emotions (angry, sadness, happy, and neutral) to improve the model's emotional expressiveness within the constraints of the dataset. Additionally, we carefully select appropriate 3D subjects for each emotion for clear emotional expression.

Our model extracts a style embedding, which encapsulates the emotional expressions conveyed in the input speech utterances. This style embedding is subsequently utilized to control the expressions of the avatar, making its facial movements more expressive even for subjects not previously seen during training. Furthermore, we are able to adjust the intensity of emotions by interpolating between style embeddings obtained from the target emotion and the neutral one. This capability enables us to generate avatars capable of displaying a broader spectrum of emotional expressions than those present solely in the existing BIWI dataset.

Our main contributions are as follows:

- To enhance facial expressions, we reclassify emotion labels into four distinct categories and employ style embeddings trained through intercross training[5].
- Our framework allows us to control the intensity of an emotion by interpolating between the target emotional style embedding and the neutral one.
- We use efficient short-term and long-term modeling approaches, such as GAU and BLSTM, to address the issue of slow inference speed.
- We show that our proposed model outperforms previous state-of-the-art models in terms of emotional expression and lip synchronization accuracy in both objective and subjective evaluation experiments.

II. Related Work

The goal of a 3D face animation model is to create a 3D avatar face from a given speech input, which can include text, speaker ID, language ID, and emotion [2], [6], [7], [11]. [6] introduce VOCA, along with the VOCASET dataset, which incorporates FLAME [11]. MeshTalk [7] is trained on an in-house dataset of 250 subjects, featuring more active facial expressions than VOCASET, including blinking and eyebrow

[1]We provide a demo page with sample videos and reclassified emotion labels at https://sam-0927.github.io/eavenet-demo/

TABLE I: Dataset of 3D facial animation.

Name	Speakers	Emotion	Duration	Release
VOCASET[6]	12	1	32m	o
Multiface[7] (subset)	13	1	21m	o
BIWI[4]	14	15	43m 35s	o
[8]	2	N/A	8m 46s	x
CREMA-D[9]	91	6	7442 clips	x
AdaIN[10]	1	8	38m 57s	x

movements. FaceFormer [2], based on the transformer [12], employs two biased attention masks and a periodic position encoding strategy. By combining FaceFormer with a pre-trained wav2vec 2.0 [13] to obtain speech representations, the model auto-regressively predicts vertices. CodeTalker [3] estimates facial movements as a discrete codebook and regressively predicts motion codes to enhance vividness and reduce the over-smoothed problem of facial movements.

[8] analyze formants and map emotional speech to 3D vertices by adding emotional embeddings to the articulation embeddings. They adopt a data-driven approach where the model automatically learns the representation of emotional states. [9] propose a model that predicts FLAME parameters to control the type and intensity of the emotion while generating expressive facial animations. During training, the model's emotion control module recognizes emotions from images, and it recognizes emotions from audio during inference. EmoTalk [14] generates emotional 3D facial animations by disentangling content and emotion in speech using cross-reconstruction loss with different emotion labels. [15] apply adaptive instance normalization (AdaIN) to separate the two aforementioned features, removing emotional information from speech and converting it to content features, including specific emotions using labels [10].

Previous studies on expressing emotions in 3D avatars have used abundant datasets containing with rich emotional information [8]–[10]. However, these datasets are often unavailable to the public, which hinders further research as described in Table I. To address this limitation, this study uses existing publicly accessible datasets.

III. Proposed Model

A. Emotion classification

The BIWI dataset [4] contains 15 emotion types and 5 scales of intensity (1:not at all $\sim$ 5:very) that indicate the level of emotional expression. Due to the small amount of training data and the uncertainty between 15 emotion types, we have decided to reclassify them into four distinct and broad categories (angry, happy, sadness, and neutral) that are still applicable to a wide range of service scenarios.

Five people were asked to re-label the emotion types using Algorithm 1 by observing facial expressions in video samples and listening to speech signals. Similar to the previous dataset, each sample was evaluated by an average of three individuals. As with the original dataset, people were asked to rate the degree of emotional expression on a scale of 1 to 5. To ensure the reliability of the labeling process and the consistency of the training data, we selected 3D subjects when the average emotional expression scores are greater than 3. For each sample of the selected subject mesh, emotion types were labeled based on the majority vote from the evaluators, which further enhances the quality of the dataset.

Algorithm 1 Emotion labeling

Input: number of samples for each 3D subject n, evaluators $m = 3$, Evaluated emotion label set $V_m^{id,index}$=$\{v_1^{1,1}, v_2^{1,1}, v_3^{1,1}, ..., v_3^{14,n}\}$, subject set S=$\{s_1, ..., s_{14}\}$, emotion set E=$\{h:0, a:0, s:0, ne:0\}$ of each sample, average expression score Scr=$\{scr_1, ..., scr_{14}\}$
Output: training subject set S_{tr}, emotion label L=$\{l^{1,1}, ..., l^{14,n}\}$

```
for i = 1 to 14 do
    if scr_i > 3 then
        S_tr ← s_i
    end if
end for
for i = 1 to 14 do
    for j = 1 to n do
        initialize values of E to 0
        for k = 1 to 3 do
            E[v_k^{i,j}] ← E[v_k^{i,j}] + 1
        end for
        l^{i,j} ← E's index with maximum value
    end for
end for
```

B. EavaNet: Emotional avatar generator

Overview. Our proposed model EavaNet, shown in Fig. 1, takes a speech signal $S_{1:N} = s_1, ..., s_N$ and a speaker ID as input, and predicts the sequences of 3D face vertices $V_{1:T} = v_1, ..., v_T$ using an encoder-decoder architecture. The encoder is composed of an audio encoder and a style encoder, which respectively predicts the contextual and style embeddings. To address the issue of insufficient training data, we use a pre-trained speech representation model, wav2vec2.0, for the audio encoder. We freeze the weights of feature extractor in this module during training. The extracted contextual embeddings are first passed through a 1D convolution layer (Conv1D). Then, they are combined with a style embedding and a speaker ID before being fed to the decoder. The Conv1D operation with a kernel size of 4, padding size of 2, and stride size of 1, performs a downsampling operation, which helps align the timing information of the acoustic features with that of the target visual vertices.

Style encoder. The style encoder computes a style embedding $e \in \mathbb{R}^{1\times d}$ that represents the facial expressions linked with each emotion from style vertices $V_{spk^*,utt^*}^{emo^{target}}$, where spk and utt denote speaker and utterance, respectively. To ensure that the style embedding exclusively encapsulates emotional information rather than contextual cues, we select style vertices solely from data within the same emotion category, excluding those that contain the target vertices ($spk^* \in S_{tr}$, $spk^* \neq spk^{target}$ and $utt^* \neq utt^{target}$, $*$ means "any"). The style encoder comprises a style extractor and a style token layer (STL), resembling the architecture of GST-Tacotron [16]. The style extractor extracts style information from the style vertices, while the STL encodes this style information into a token that is then used by the decoder to generate the output. The style extractor is made up of a projection layer,

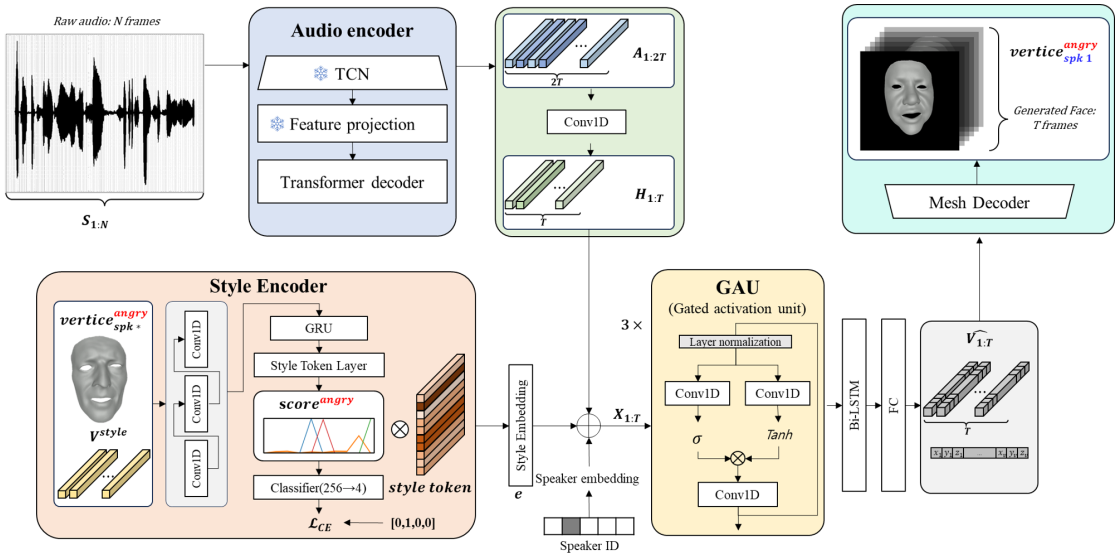

Fig. 1: Architecture of EavaNet. The snow symbols in the audio encoder denote freezing the weights.

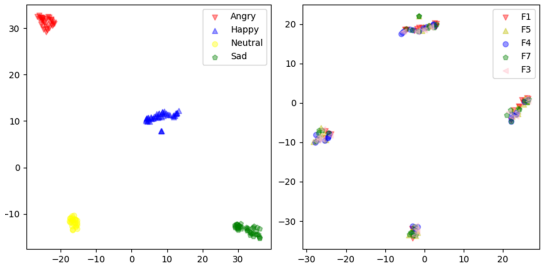

Fig. 2: t-SNE [18] plots of style embeddings from five subjects in training data (left: emotion, right: speaker).

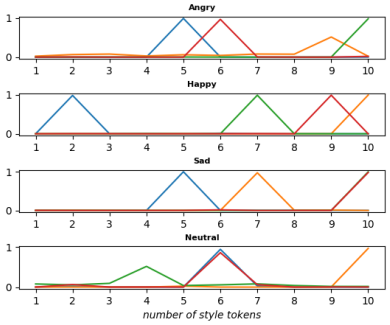

Fig. 3: The score of 4-head attention in the style token layer. Each color represents the index of each head, and each emotion has a different distribution of style tokens.

three Conv1D layers with layer normalization, LeakyReLU activation functions, and a bidirectional gated recurrent unit (GRU) [17]. The projection layer transforms the input data into a feature vector. The Conv1D layers, with a channel size of 256 and kernel size of 4, along with a stride of 2 and padding size of 1, extract features from the feature vector. Subsequently, layer normalization is applied to normalize these features. The GRU then processes the features to extract a 256-dimensional style embedding.

$$\mathbf{e} = \text{STL}(\text{style}_{\text{extractor}}(V^{emo^{target}}_{spk^*,utt^*})). \quad (1)$$

In the 4-head attention layer of STL, the output of the style extractor is multiplied by the style tokens $V_{style\,token}$ and each token's score. The attention score is calculated by multiplying the query Q from the output of the style extractor with the transposed trainable parameter key K^T, then dividing it by the square root of the key's dimension d_k. Finally, the weighted sum of style tokens for each head is concatenated to form the style embedding.

$$\text{Att}(Q, K, V_{style\,token}) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V_{style\,token}. \quad (2)$$

To separate the style embeddings depending on the type of emotion, we compute the cross-entropy $\mathcal{L}_{CE}$ between the style embedding and the emotion labels rather than using contrastive learning technique in embedding space [19]. This allows us to identify the different styles that are associated with each emotion, as shown in Fig. 2. In the figure, style embeddings that belong to the same emotion category become closer to each other, forming independent clusters regardless of the speaker variations. This means that the style of speech can be used to identify the emotion expressed, even if the speaker's voice is different. The style embedding characteristics can be determined by the score since the style tokens are shared for all training data.

As shown in Fig. 3, the score of each emotion highlights different style tokens. This means that the style embedding can be represented by the combination of style tokens, as they are tailored to their specific characteristics. In other words, the style embedding can be created by combining different style tokens, each of which represents a specific emotion. To use the style embeddings during inference, we first compute the centroid of the generated style embeddings in each emotion cluster, then store them in a lookup table during training. Our model directly retrieves the style embeddings from the pre-trained lookup table during inference.

Decoder. The decoder is composed of GAU, BLSTM and

projection layers motivated by [20]. The style embedding and speaker embedding $e_{spk} \in \mathbb{R}^{1\times d}$ are added to the hidden representations of the Conv1D layers and they are passed to the GAU.

$$\begin{aligned} \mathbf{X_{1:T}} = \mathbf{e} \oplus \mathbf{H_{1:T}} \oplus \mathbf{e_{spk}}, \qquad \mathbf{e_{spk}} = F_\theta(i), \\ \mathbf{H_{1:T}} = \text{Conv1D}(\mathbf{A_{1:2T}}), \\ \mathbf{A_{1:2T}} = \text{Enc}_{\text{audio}}(\mathbf{S_{1:N}}), \end{aligned} \tag{3}$$

where $X \in \mathbb{R}^{T\times d}$ denotes the input of GAU and $\oplus$ means adding. $H \in \mathbb{R}^{T\times d}$ and $A \in \mathbb{R}^{2T\times d_A}$ are hidden representations and audio encoder outputs, respectively. F_θ and i depict fully connected layers and a speaker ID, respectively. The GAU used in the generative model [21] has three residual blocks, each consisting of two paths: a filter and a gate, with residual connections. Each path has a single 1D convolution layer and two activation functions (hyperbolic tangent and sigmoid). The filter path estimates local features in adjacent frames based on the receptive field size, and the gate path determines how much to pass through. After the outputs of the two gates are multiplied, they are passed through an output 1D convolution layer and are added to the residual connections, which have 256 channel size with 1 kernel size. The convolution layers have channel sizes of (256, 512) and kernel sizes of (5, 3, 5) with padding sizes of (2, 1, 2).

$$\begin{aligned} z_k = \tanh(W_{f,k} * x_k + b_{f,k}) \otimes \sigma(W_{g,k} * x_k + b_{g,k}), \\ x_{k+1} = (W_{o,k} * z_k + b_{o,k}) \oplus x_k, \end{aligned} \tag{4}$$

where tanh, σ, $*$ and $\otimes$ denote hyperbolic tangent, sigmoid, convolution operator, and an element-wise multiplication operator, respectively. The f, g and, o denote filter, gate, and output layer, respectively. $W_{*,k}$ and $b_{*,k}$ are trainable weights and the bias of the kth layer.

All of these processes are conducted concurrently. Since the GAU uses only CNNs to capture local features, we also incorporate two BLSTM layers with a channel size of 256 to capture temporal characteristics. Finally, the output of the BLSTM layers is projected to the vertices:

$$\hat{V}_{1:T} = P_\theta(\text{BLSTM}(\mathbf{Y}_{1:T})), \tag{5}$$

where $Y_{1:T}$ and $\hat{V}_{1:T}$ are the outputs of GAU and predicted vertices from fully connected layers P_θ, respectively.

Training criterion and details. We compute mean square errors (MSEs) between the predicted vertices $\hat{V}_{1:T}$ and the target ones $V_{1:T}$.

$$\mathcal{L}_{\text{vert}} = \frac{1}{T}\sum_{i=1}^{T} \|V_i - \hat{V}_i\|_2, \quad V_i^j = V_i^j - m^j, \tag{6}$$

where j represents subject id. The total training loss is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{vert}} + \lambda_{CE}\mathcal{L}_{\text{CE}}, \tag{7}$$

where the scale factor λ_{CE} is set to 0.001.

We trained EavaNet for 200 epochs on an NVIDIA GeForce RTX 3090 GPU using the Adam optimizer. The initial learning rate is set to 1e-4 and is reduced by half every 40 epochs. The mini-batch size is set to 1. For the reference models for comparison, we adhere to their respective official codes for training.

TABLE II: Quantitative evaluation on BIWI-A.

Model	Vertex error ↓ ($\times 10^{-5}$mm)	Lip vertex error ↓ ($\times 10^{-4}$mm)	FDD ↓ ($\times 10^{-5}$mm)
FaceFormer [2]	3.9876	7.5916	5.0914
CodeTalker [3]	4.7354	8.2971	**3.6702**
EavaNet	**3.4603**	**6.3628**	3.9958

TABLE III: A/B results of EavaNet and previous models.

Model		Lip sync↑	Expression↑
EavaNet	vs GT	40.86	45.21
	vs FaceFormer	51.73	66.23
	vs CodeTalker	53.52	60.41

IV. Experiments

A. Experiment setting

We use the BIWI dataset [4], which contains emotional speech accompanied by dense dynamic 3D facial geometries. The dataset consists of 14 subjects, each with 40 sentences, for an average sequence length of 4.67 seconds. The 3D facial meshes are captured at a rate of 25 frames per second, with each mesh consisting of 23,370 vertices. We eliminate outlier samples by following the BIWI guidelines; instances with zero scores across all 15 emotion categories are considered outliers that should be removed. The dataset is divided into three parts: a training set with 146 sentences spoken by five speakers (F1, F3, F4, F5, F7); a validation set with 18 sentences spoken by the same five speakers; and two separate test sets, BIWI-A and BIWI-B. BIWI-A has 19 sentences spoken by five subjects who were previously seen, while BIWI-B has 30 sentences spoken by nine subjects who were not seen before. We use BIWI-A for objective evaluation, and BIWI-B for user studies and ablation studies.

To stabilize the training, we removed the silence sections before and after the main audio segments using the *librosa* library, aligned the video accordingly, and normalized the audio scale to -3dB. The audio samples were captured at a rate of 16,000 samples per second.

B. Quantitative Evaluation

To measure the performance of our model, we use two metrics: lip vertex error and the *upper-face dynamics deviation* (FDD), following the methodologies of FaceFormer and CodeTalker. The lip vertex error measures the accuracy of lip-syncing, while the FDD assesses the fidelity of facial expressions. Both of these metrics measure the discrepancy between the predicted and actual vertices. The lip vertex error is a measure of the maximum distance between the actual and predicted lip positions for each frame, averaged over all frames. The FDD measures the difference in standard deviation between the vertices of ground-truth and predicted facial expressions for each frame, which are associated with upper facial expressions. Furthermore, we compute the L2 error for all vertices in each frame and then average these

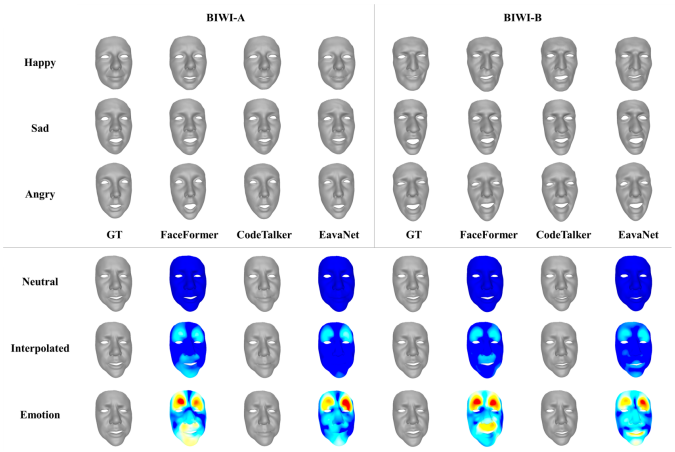

Fig. 4: Visual comparison of sampled 3D faces generated by different models (top). 3D faces and their heatmaps animated by EavaNet with neutral, interpolated and, emotional style embedding (bottom). We use the *e38* sentence in BIWI-B (*angry* emotion) in this sample.

values across all frames. These metrics are used to fully assess the overall quality of the generated 3D avatars.

Table II presents the quantitative findings for BIWI-A. Our model outperforms previous state-of-the-art models in both lip accuracy and overall quality. This result highlights the fact that our model can accurately generate 3D facial animations while conveying distinct pronunciations. Our model also has a better FDD score than FaceFormer, which indicates its superiority in facial expression representations. While CodeTalker outperforms EavaNet in terms of FDD score, the overall quality of its generated avatars is inferior to that of other models.

C. Qualitative Evaluation

Our model's ability to create expressive 3D facial animation with precise lip synchronization has been confirmed through qualitative evaluation. A visual comparison of EavaNet and competing models is presented in Fig. 4. To ensure a fair comparison, we use the same randomly selected subject ID as the conditional input for FaceFormer, CodeTalker, and our proposed model. In both cases, the samples produced by EavaNet express the emotion more clearly through dynamic facial movements, while other samples do not, especially with drooping eye tails and wrinkles between the brows when expressing *sadness*. One interesting finding is that our model expresses *sadness* and *angry* more strongly than the ground truth (GT) in both BIWI-A and BIWI-B (the 2nd and 3rd rows of Fig. 4). Unlike other models, EavaNet can successfully convey emotions even when they cannot be discerned solely from the prosody of the speech input because it uses style embeddings effectively. We suggest that you zoom in and examine the figure more closely.

D. User study

We conducted preference A/B tests with GT, FaceFormer, and CodeTalker to evaluate the quality of the generated avatars from a human perspective. Twenty-three people were asked to choose samples with high lip-sync accuracy from the outputs of two different models. In addition, participants were asked to choose samples that effectively conveyed the emotions expressed in the audio input. We created avatars for unseen subjects in BIWI-B by using five seen subjects' IDs as conditions, and then picked 30 samples from all the conditioned IDs. Using this setup, we created 90 A/B pairs (30 samples $\times$ 3 comparison models).

Table III demonstrates a considerable improvement in our model's emotional expression and lip synchronization accuracy with audio. Our approach has been shown to be more accurate in terms of lip synchronization and emotional expression than state-of-the-art models, with a preference of over 50% and 60%, respectively. In particular, our model demonstrates a 66% preference in expressing emotions compared to avatars generated by FaceFormer, and it outperforms CodeTalker, which uses vector quantization (VQ) to address smoothing issues, by 60%. Additionally, EavaNet performs similarly to the ground truth (GT), with a preference of 45%. This implies that by training the model on the five subjects with the richest emotional expression from the 14 subjects in the BIWI dataset (as mentioned in Section III-A), we can effectively overcome limitations and create authentic facial expressions, even for subjects with lower emotional expression.

TABLE IV: Results in terms of inference speed.

State	FaceFormer	CodeTalker	EavaNet
RTF	0.1356	1.5684	**0.0097**
Ratio	$\times$ 11.56	$\times$ 1	$\times$ **161.55**

TABLE V: MOS (mean opinion score) results with 95% confidence intervals in terms of emotion manipulation.

State	Neutral	Interpolated	Emotion
MOS	2.44$\pm$0.29	3.13$\pm$0.21	4.11$\pm$0.19

E. Inference speed

When deploying the model in real-world applications, it is essential to generate high-quality 3D facial animations with human-like emotional expressions in a time-efficient manner. To assess the generation speed of each model, we randomly selected a sample from the test set and measured the generation time using an NVIDIA GeForce RTX 3090 GPU. We calculated the average time required after conducting 100 synthesis operations. The RTF (Real Time Factor) denotes the ratio of the time taken to predict the positions of 3D vertices for a 4-seconds length sequence.

As detailed in Table IV, our proposed model demonstrates the capability to generate 3D avatars at speeds 14 and 161 times faster than FaceFormer and CodeTalker, respectively. FaceFormer and CodeTalker rely on autoregressive (AR) modeling, wherein predictions of vertices are conditioned on previous predicted frames. Consequently, these models demand a significant amount of time during inference due to the sequential nature of their predictions. In contrast, EavaNet adopts a non-AR structure that enhances inference speed by enabling parallel generation of vertices. Although the model incorporates BLSTM layers after GAUs, it processes data in latent space sequentially, mitigating computational overhead.

F. Ablation study

Emotion intensity. Style embeddings are beneficial for producing high-quality 3D avatars and controlling the intensity of emotional expression. We use linear interpolation to blend the style embedding between neutral and (happy, sadness, anger) to control the intensity of the expression.

$$e_{\alpha}^{emotion} = \alpha e^{emotion} + (1-\alpha)e^{neutral}, \tag{8}$$

where $e^{emotion}$ and $e^{neutral}$ are style embeddings extracted from the trained lookup table (see Section III-B). We set the ratio α to 0.3 to perform exemplary experiments. We randomly selected 45 samples (15 samples $\times$ 3 expression strength; neutral, weak emotion, and emotion) from BIWI-B by conditioning the subject *F4*, and then asked 25 people to rate the strength of emotional expression on a scale of 1 to 5. The higher the score, the more intense the emotional expression. As shown in Table V, the avatar generated using the interpolated style embedding $e_{\alpha}^{emotion}$ has a weaker emotional representation than those generated using the emotion style embedding $e^{emotion}$, but a stronger emotional representation than those generated using the neutral style embedding $e^{neutral}$. Fig. 4 shows an example. The wrinkles between the brows become more pronounced and the facial expression changes as the emotion of anger intensifies, gradually strengthening the emotional expression.

V. Conclusion

In this work, we presented EavaNet, a generative model that creates 3D facial animations with emotional expression given speech signals. EavaNet is non-autoregressive, and is able to quickly predict 3D face vertices showing accurate emotional facial expressions. To address limited training data, we re-categorized BIWI dataset labels into four emotions. Additionally, EavaNet estimates style embeddings and interpolates between them enables fine-tuned control over emotional expression strength. Our model outperforms previous state-of-the-art models in terms of emotion expression and lip sync accuracy in both subjective and objective evaluations.

References

[1] H. X. Pham, S. Cheung, and V. Pavlovic, "Speech-driven 3d facial animation with implicit emotional awareness: A deep learning approach," in *Proc. CVPR*, 2017.

[2] Y. Fan, Z. Lin, J. Saito, W. Wang, and T. Komura, "Faceformer: Speech-driven 3d facial animation with transformers," in *Proc. CVPR*, 2022.

[3] J. Xing, M. Xia, Y. Zhang, X. Cun, J. Wang, and T.-T. Wong, "Codetalker: Speech-driven 3d facial animation with discrete motion prior," in *Proc. CVPR*, 2023.

[4] G. Fanelli, J. Gall, H. Romsdorfer, T. Weise, and L. Van Gool, "A 3-d audio-visual corpus of affective communication," *Transactions on Multimedia*, vol. 12, 2010.

[5] Y. Bian, C. Chen, Y. Kang, and Z. Pan, "Multi-reference tacotron by intercross training for style disentangling, transfer and control in speech synthesis," in *interspeech*, ISCA, 2019.

[6] D. Cudeiro, T. Bolkart, C. Laidlaw, A. Ranjan, and M. J. Black, "Capture, learning, and synthesis of 3d speaking styles," in *Proc. CVPR*, 2019.

[7] A. Richard, M. Zollhöfer, Y. Wen, F. De la Torre, and Y. Sheikh, "Meshtalk: 3d face animation from speech using cross-modality disentanglement," in *Proc. CVPR*, 2021.

[8] T. Karras, T. Aila, S. Laine, A. Herva, and J. Lehtinen, "Audio-driven facial animation by joint end-to-end learning of pose and emotion," *ACM Transactions on Graphics*, vol. 36, 2017.

[9] Y. Chen, J. Zhao, and W.-Q. Zhang, "Expressive speech-driven facial animation with controllable emotions," *Proc. ICMEW*, 2023.

[10] C.-J. Chang, L. Zhao, S. Zhang, and M. Kapadia, "Disentangling audio content and emotion with adaptive instance normalization for expressive facial animation synthesis," *Computer Animation and Virtual Worlds*, vol. 33, 2022.

[11] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero, "Learning a model of facial shape and expression from 4d scans.," *ACM Transactions on Graphics*, vol. 36, 2017.

[12] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Proc. NIPS*, vol. 30, 2017.

[13] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," *Proc. NIPS*, vol. 33, 2020.

[14] Z. Peng, H. Wu, Z. Song, *et al.*, "Emotalk: Speech-driven emotional disentanglement for 3d face animation," *Proc. ICCV*, 2023.

[15] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. ICCV*, 2017.

[16] Y. Wang, D. Stanton, Y. Zhang, *et al.*, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *Proc. ICML*, 2018.

[17] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proc. NIPS*, 2014.

[18] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, 2008.

[19] J. Kim and H.-G. Kang, "Contrastive learning based deep latent masking for music source separation," in *Proc. ISCA*, 2023.

[20] D. Kim, S.-W. Chung, H. Han, Y. Ji, and H.-G. Kang, "Hd-demucs: General speech restoration with heterogeneous decoders," in *Proc. ISCA*, 2023.

[21] A. van den Oord, S. Dieleman, H. Zen, *et al.*, "Wavenet: A generative model for raw audio," in *Proc. ISCA*, 2016.