EmoDiffTalk: Emotion-aware Diffusion for Editable 3D Gaussian Talking Head

Supplementary Material

1. More Details for Canonical Gaussian Rig Reconstruction

Canonical Gaussian Rig Reconstruction. We use the OTF points designed by Emotalk3D [3] to perform canonical Gaussian rig reconstruction. Using such OTF mechanism, Gaussian points in non-facial regions (e.g., clothing, hair) exhibit minimal subtle changes during speech, suggesting that these points should be determined after establishing the canonical appearance and then driven structurally by facial motion (without updating parameters via backpropagation during the dynamic process). Besides, the total number of Gaussian points are also fixed (as the vertices number of template mesh) used to represent these regions.

Specifically, we use the provided mesh template by Emotalk3D, and bind Gaussian points to each vertex to represent the facial region, while the quantity of Gaussian points for other parts is predetermined. During the training phase at this stage, we learn non-facial regions' points positions and parameters, as well as the attributes (excluding position) of the facial region points. For all of the datasets we used, we conduct the similar canonical Gaussian rig reconstruction for thereafter 3D Gaussian talking head editing.

Canonical Appearance via Triplane. In traditional 3DGS, each 3D Gaussian is represented by the 3D point position μ and covariance matrix Σ, and the density function is formulated as:

$$g(x) = e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}. \tag{1}$$

As 3D Gaussians can be formulated as a 3D ellipsoid, the covariance matrix Σ is further formulated as:

$$\Sigma = RSS^TR^T, \tag{2}$$

where S is a scale and R is a rotation matrix. The 3D Gaussians are differentiable and can be easily projected to 2D splats for rendering.

For the appearance modeling, 48 out of the 59 parameters in each Gaussian distribution are used for SH (3rd-order) to capture viewpoint-dependent color. Recently, methods using triplanes [9] to store features and decoding color via MLP have demonstrated superiority over traditional representations [5]. For Gaussian talking heads, the generation process typically involves minimal strong light changes, with variations primarily manifesting in facial muscle details. These details have been demonstrated to be effectively simulated by opacity variations. Therefore, storing the space-intensive SH is unnecessary.

Different from the original 3DGS that uses spherical harmonics for appearance modeling, we employ a triplane representation combined with MLP decoding for color generation. Specifically, each Gaussian point queries features from three orthogonal feature planes (XY, XZ, and YZ planes) and decodes them through an MLP network to produce the final RGB color. In the differentiable rendering phase, $g(x)$ is multiplied by an opacity α, then splatted onto 2D planes and blended to constitute colors for each pixel. Different from the original 3DGS that uses spherical harmonics for appearance modeling, we employ a triplane representation to generate the initial RGB color for each Gaussian point during the canonical stage. Specifically, the color c for each point is generated by:

$$c = \mathcal{M}(F_{xy}(x,y) \oplus F_{xz}(x,z) \oplus F_{yz}(y,z)), \tag{3}$$

where $\mathcal{M}$ is an MLP decoder and F_{xy}, F_{xz}, F_{yz} are triplane feature maps.

Once the canonical Gaussians are established, we store the precomputed RGB color c directly. During animation, the color remains static while only the opacity undergoes dynamic changes. This design ensures color consistency while allowing expressive motion variations.

In this way, the appearance of a static head can be represented as 3D Gaussians G:

$$G \leftarrow \{\mu, S, R, \alpha, c\}, \tag{4}$$

where $\leftarrow$ means G is a set of parameterized points, each represented by a parameter set on the right of the arrow, and c is the precomputed RGB color generated from the triplane representation.

In our approach, the canonical 3D Gaussians G_0 represent a static head avatar and are learned from multi-view images of a moment without speech, usually the first frame of a video clip. The canonical 3D Gaussians G_0 are denoted as:

$$G_0 \leftarrow \{\mu_0, S_0, R_0, \alpha_0, c_0\}. \tag{5}$$

2. More Details for AU-prompt Gaussian Diffusion

Speech-to-AU Transformer Encoder. We map audio to structured AU representations via Speech-to-AU Encoder. The structure of the encoder is shown in Fig. 1 (left), which adopts a three-stage cascaded architecture. First, the HuBERT [4] model is utilized to extract temporal representational features from the audio input.The feature sequence is denoted as:

$$\boldsymbol{A}_{0:T-1} = \{\boldsymbol{A}_t \mid t = 0, \ldots, T-1\}, \quad \boldsymbol{A}_t \in \mathbb{R}^{768}. \tag{6}$$

Then, a Transformer Encoder captures long-range dependencies within the sequence. Lower layers use constrained

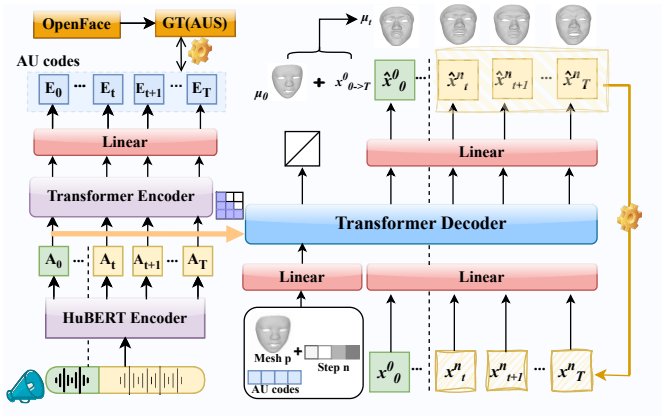

Figure 1. Speech-to-AU Encoder pipeline (left) and AU-prompt Gaussian Diffusion pipeline(right).

attention to capture rapid articulatory changes (e.g., lip closure), while upper layers model slower prosodic variations. A residual projection head maps hidden states to a calibrated AU space for subsequent text-based modulation. After that, a fully connected layer projects the high-dimensional features into a 17-dimensional AU coding space, corresponding to the intensity of 17 target Action Units. Finally, a lightweight frame-pooling layer aligns AU features with facial frame rates, outputting temporally aligned AU Codes:

$$\boldsymbol{E}_{0:T-1} = \{\boldsymbol{E}_t \mid t = 0, \ldots, T-1\}, \quad \boldsymbol{E}_t \in \mathbb{R}^{17}. \tag{7}$$

During the training phase, we employ the OpenFace [1] toolkit to process the original video frames to obtain the ground truth (GT):

$$\boldsymbol{a}_{0:T-1} = \{\boldsymbol{a}_t \mid t = 0, \ldots, T-1\}, \quad \boldsymbol{a}_t \in \mathbb{R}^{17}, \tag{8}$$

which serves as the supervisory signal for model parameter optimization. And the encoder is trained with a combination of regression and temporal consistency losses:

$$\mathcal{L}_{\text{reg}} = \sum_{t=0}^{T} \sum_{k=1}^{K} \text{L1}(E_{t,k} - \hat{a}_{t,k}), \tag{9}$$

$$\mathcal{L}_{\text{temp}} = \sum_{t=1}^{T} \left\| E_t - E_{t-1} \right\|^2, \tag{10}$$

$$\mathcal{L}_{\text{audio}} = \lambda_{\text{reg}} \mathcal{L}_{\text{reg}} + \lambda_{\text{temp}} \mathcal{L}_{\text{temp}}, \tag{11}$$

where $\lambda_{\text{reg}} = 1.0$ and $\lambda_{\text{temp}} = 0.1$ balance the regression accuracy and temporal smoothness.

AU-prompt Gaussian Diffusion. This module incorporates the AU Codes obtained from the Speech-to-AU encoder into facial motion modeling.

The forward process follows a Markov chain $q(\boldsymbol{x}_t^n \mid \boldsymbol{x}^{n-1} * t)$ for $n \in \{1, \ldots, N\}$ that gradually adds Gaussian noise to the original facial motion sequence $\boldsymbol{x}_t^0$ according to a predefined variance schedule, ultimately transforming it into a standard normal distribution. The reverse process reconstructs the original sequence by learning the distribution $q(\boldsymbol{x}^{n-1}t \mid \boldsymbol{x}^n t)$.

Specifically, we utilize AU Codes $\boldsymbol{E}_{0:T-1}$ extracted from audio features as one input to guide the denoising process. Second, we employ mesh point positions $\boldsymbol{P}$ as the initial template, with the network learning the offsets of all mesh points $\Delta \boldsymbol{P}_t$. And we also use Hubert-extracted audio features as one of its inputs and incorporate a windowing mechanism. Naturally, learning point offsets presents greater challenges than predicting prior 3DMM [2] coefficients. Therefore, we designed a series of losses to constrain facial structural changes including:

$$\mathcal{L}_{\text{vertex}} = \frac{1}{T \cdot V} \sum_{t=0}^{T-1} \sum_{v=0}^{V-1} \|x_t^0(v) - \hat{x}_t^0(v)\|^2, \tag{12}$$

$$\mathcal{L}_{\text{vel}} = \frac{1}{T \cdot V} \sum_{t=0}^{T-1} \sum_{v=0}^{V-1} \|\nabla x_t^0(v) - \nabla \hat{x}_t^0(v)\|^2, \tag{13}$$

$$\mathcal{L}_{\text{acc}} = \frac{1}{(T-1) \cdot V} \sum_{t=0}^{T-2} \sum_{v=0}^{V-1} \|\nabla^2 x_t^0(v) - \nabla^2 \hat{x}_t^0(v)\|^2, \tag{14}$$

$$\mathcal{L}_{\text{motion}} = \mathcal{L}_{\text{vel}} + \mathcal{L}_{\text{acc}}, \tag{15}$$

$$\mathcal{L}_{\text{lip}} = \frac{1}{T_{\text{lip}} \cdot V_{\text{lip}}} \sum_{t \in \mathcal{T}_{\text{lip}}} \sum_{v \in \mathcal{V}_{\text{lip}}} \|x_t^0(v) - \hat{x}_t^0(v)\|^2. \tag{16}$$

The total geometry loss combines these components with deformation regularization:

$$\begin{aligned} \mathcal{L}_{\text{geometry}} = \lambda_{\text{vertex}} \mathcal{L}_{\text{vertex}} + \lambda_{\text{motion}} \mathcal{L}_{\text{motion}} \\ + \lambda_{\text{deform}} \mathcal{L}_{\text{deform}} + \lambda_{\text{lip}} \mathcal{L}_{\text{lip}}, \end{aligned} \tag{17}$$

where $\lambda_{\text{vertex}} = 1.0$, $\lambda_{\text{motion}} = 0.5$, $\lambda_{\text{deform}} = 0.1$, and $\lambda_{\text{lip}} = 0.8$. The denoising network then outputs the clean sequence as:

$$\hat{\boldsymbol{x}}_{0:T} = D\theta(\boldsymbol{x}^n 0 : T, \boldsymbol{P}, \boldsymbol{E}_{0:T}, \boldsymbol{A}_{0:T}, n). \tag{18}$$

Feature line [8]. For opacity modeling, we designed a compact Feature Line whose initial purpose was to store implicit features regarding variations in flame expression coefficients[ ]. The Feature Line regularization $\mathcal{L}_{\text{reg}}$:

$$\mathcal{L}_{\text{reg}} = \lambda_{\text{sparse}} \|\mathcal{F}\|_1 + \lambda_{\text{smooth}} \sum_{k=1}^{K-1} \|\mathbf{f}_k - \mathbf{f}_{k+1}\|_2^2, \tag{19}$$

with $\lambda_{\text{sparse}} = 0.01$ and $\lambda_{\text{smooth}} = 0.001$.

Here, we repurpose it to store fine-grained features of AU Codes changes related to opacity. The learnable feature line $\mathcal{F} \in \mathbb{R}^{17 \times Q \times 16}$ captures AU-specific opacity patterns, where Q is the number of facial Gaussian points. This continuous AU-based representation enables smooth expression

interpolation and infinite blending possibilities.
where $\mathbf{f}_t^i$ is the feature combination weighted by AU intensities.
We enforce motion-opacity correlation to ensure physical plausibility: regions with larger geometric deformations exhibit proportional opacity changes, maintaining consistency between motion and appearance variations including:

$$\mathcal{L}_{\text{recon}} = (1 - \lambda_{\text{ssim}})\mathcal{L}_1 + \lambda_{\text{ssim}}\mathcal{L}_{\text{ssim}}, \tag{20}$$

$$\mathcal{L}_{\text{opcmotion}} = \lambda_{\text{opcmotion}} \sum_{i=1}^{Q} \left(\|\Delta\mu_t^i\|_2 - \gamma \cdot |\Delta\alpha_t^i| \right)^2, \tag{21}$$

$$\mathcal{L}_{\text{dist}} = \lambda_{\text{move}} \sum_{i=1}^{Q} \min\left(\|\Delta\mu_t^i\|_2, \tau \right). \tag{22}$$

3. More Details for Text-to-AU Emotion Controller

Dataset. We leverage the powerful language analysis capabilities of GPT-5 to enhance our model's ability to perform expression analysis on an open vocabulary. The prompt used is as follows:

Acting as an expert proficient in FACS (Facial Action Coding System), please generate a JSON dataset containing 100 entries, strictly adhering to the specified sequence of 17 Action Units (AUs): ["AU01", "AU02", "AU04", "AU05", "AU06", "AU07", "AU09", "AU10", "AU12", "AU14", "AU15", "AU17", "AU23", "AU24", "AU25", "AU26", "AU45"]. Each entry must be clearly categorized as either a simple expression (e.g.,a person is winking), a complex expression combination (e.g., A person with a concerned frown and raised inner brows.), or a basic/mixed emotion (e.g.,an angry person,a sad-relieved person), ensuring all AU combinations comply with facial muscle movement logic. The output must be structured JSON, including precise AU labels (a 0/1 vector), a core description, three positive samples with high semantic/visual correlation, and three clearly contrasting negative sample descriptions. Please apply a phased processing strategy: first, plan the anatomical plausibility of the AU combinations, then generate the descriptions and verify the distinctiveness between positive and negative samples, and finally, output uniformly to ensure consistency in linguistic diversity and logic across the dataset.

We have manually reviewed each data entry and removed those of poor quality, resulting in a final dataset of 350 high-quality samples containing facial expressions and actions. Table 1 shows some examples of the data.

Train. During training we use two loss functions: the weighted Focal binary cross-entropy loss and an improved InfoNCE contrastive loss, where the weighted Focal BCE is

$$\mathcal{L}_{\text{BCE}} = \frac{1}{N} \sum_{i=1}^{N} \Big[-\alpha\, y_i (1 - p_i)^{\gamma} \log p_i - (1 - \alpha)(1 - y_i) p_i^{\gamma} \log(1 - p_i) \Big], \tag{23}$$

with $y_i \in \{0, 1\}$ the ground-truth label and $p_i = \sigma(\text{logit}_i)$ the predicted probability. We set the hyperparameters $\alpha = 0.35$ and $\gamma = 3.0$. This loss is the main classification objective: α balances positive and negative samples, and the factor $(1-p)^{\gamma}$ up-weights hard-to-classify examples to improve discrimination for rare or difficult AUs.

The improved InfoNCE structured contrastive loss is

$$\mathcal{L}_{\text{infoNCE}} = -\log \frac{\exp(\text{sim}(z_a, z_p)/\tau)}{\exp(\text{sim}(z_a, z_p)/\tau) + \sum_n \exp(\text{sim}(z_a, z_n)/\tau)}, \tag{24}$$

where z_a, z_p, z_n denote the feature vectors of anchor, positive and negative samples (features are typically L_2 normalized), $\text{sim}(\cdot, \cdot)$ is the similarity (dot product after normalization), and the temperature is $\tau = 0.07$. This auxiliary loss pulls semantically matching text and AU features closer and pushes different-semantic features apart, helping to reduce the semantic gap between CLIP [7] text features and AU-specific features. The total loss is a weighted sum:

$$\mathcal{L} = \lambda_{\text{BCE}}\mathcal{L}_{\text{BCE}} + \lambda_{\text{infoNCE}}\mathcal{L}_{\text{infoNCE}}, \tag{25}$$

and we set the weighted Focal BCE weight $\lambda_{\text{BCE}} = 01$ and the contrastive weight $\lambda_{\text{infoNCE}} = 0.005$ so the classification task remains dominant while contrastive learning provides a helpful auxiliary signal. The model is optimized with AdamW [6] using a learning rate of 1×10^{-4}, weight decay of 0.01, and betas $(\beta_1, \beta_2) = (0.9, 0.999)$. Training runs for 300 epochs with a batch size of 128.

Evaluation. To quantitatively evaluate the performance of the proposed Text-to-AU Emotion Controller, we employed a comprehensive set of metrics, including Accuracy, F1-score, Precision, and Recall. This multi-faceted evaluation provides a holistic view of the model's classification capability and its robustness across various Action Units.
Our model demonstrates exceptional performance on the test set, achieving an overall accuracy of 0.9753. More importantly, the model attains a mean F1-score of 0.9030, balancing a high mean precision of 0.8958 and an outstanding mean recall of 0.9386 . These results indicate that the model is not only highly accurate in its predictions but also excels at identifying the majority of relevant AU activations with minimal false negatives.

4. Details of User Study

As described in the main paper, we recruited 35 volunteers to evaluate videos generated by different methods across four

Table 1. Examples of text-AU pair generated by GPT-5

Description	Type	activated AUs
A sad person	Emotion	AU01, AU02, AU04, AU15
A happy person	Emotion	AU06, AU12
A happy-surprised person	Emotion	AU01, AU02, AU05, AU06, AU12, AU25
A person with raised eyebrows	Expression	AU01, AU02
A person with concerned frown	Expression	AU01, AU04
An incredulous person with raised brows and open mouth	Expression	AU01, AU02, AU24, AU25

dimensions: Speech-Visual Synchronization (S-V Sync.), Video Fidelity, Image Quality, and Emotion Control. Participants rated each video on a 1-5 scale, with the final score being the mean of all 35 ratings.

Here we provide the complete questionnaire design and detailed assessment criteria used in our user study.

4.1. Questionnaire Design

For each video clip, participants were asked to answer the following questions and provide ratings on a 5-point Likert scale (1 = Very Poor, 5 = Excellent):

Q1. Speech-Visual Synchronization: *"How well do the lip movements synchronize with the audio speech content?"*

Q2. Video Fidelity: *"How realistic and temporally coherent is the overall video? Does it appear indistinguishable from real videos?"*

Q3. Image Quality: *"How would you rate the visual quality of the generated frames in terms of sharpness, clarity, and absence of artifacts?"*

Q4. Emotion Control*: *"How accurately and naturally does the facial expression reflect the intended emotion described in the text prompt?"*

**Note: Q4 was only evaluated for methods supporting text-based emotion control (i.e., Hallo3 and our method).*

4.2. Detailed Assessment Criteria

To ensure consistency across participants, we provided the following detailed criteria for each dimension:

- **Speech-Visual Synchronization:** This indicator evaluates the temporal alignment between lip movements and audio speech, focusing on phoneme-viseme correspondence. A score of 5 indicates perfect synchronization with no perceivable delay, while a score of 1 indicates severe misalignment where lip movements are completely out of sync with the audio.
- **Video Fidelity:** This indicator assesses whether the generated video is vivid, natural, and temporally coherent, appearing indistinguishable from real recordings. Higher scores indicate smoother frame transitions, absence of flickering or jittering, and overall realism in facial dynamics.
- **Image Quality:** This indicator evaluates the per-frame visual quality as perceived by the human vision system. Specifically, videos with higher clarity, sharper details, better color fidelity, and fewer compression artifacts or rendering defects receive higher scores.
- **Emotion Control:** This indicator measures how accurately the generated facial expression matches the intended emotion specified in the text prompt, and whether the expression appears natural and appropriately intense. Higher scores indicate better alignment between the text description and the visual emotional expression.

4.3. Evaluation Procedure

Participants were shown videos in randomized order to avoid bias. For the EmoTalk3D dataset, 5 video sequences with different identities were selected; for the RenderMe-360 dataset, 2 sequences were used. Participants could replay videos as needed before providing their ratings.

5. Potential Failure Cases

As illustrated in **Fig.**2, we identified three primary failure cases, including (a) reconstruction from view occlusions, (b) fast head movements (like the hairs), and (c) extreme expressions (like the eye rollments) respectively. How to further improve the generation quality from those challenging cases is an interesting direction, which we leave for future works.

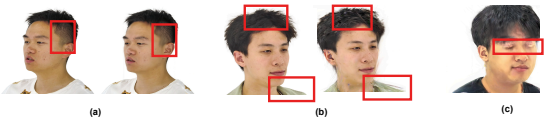

Figure 2. Visual results of failure cases.

6. More Analysis of AU Prediction Accuracy and Motion Precision

To further evaluate audio-to-AU prediction accuracy, we measured the fine-grained prediction capability of AU codes (range 0-5), achieving an F1-Score of **0.86** and an accuracy of **95.13%** (with MAE < 0.01). This demonstrates that EmoDiffTalk maintains a high degree of robustness against

Table 2. Analysis of AU Prediction Accuracy and Motion Precision

(a) Encoder Performance		
Metric	**F1-Score ↑**	**Acc (MAE<0.01) ↑**
Ours	**0.86**	**95.13%**

(b) Decoder Comparison			
Metric	**LVE (mm) ↓**	**FDD (10^{-5}m) ↓**	**MOD (mm) ↓**
S2GNet (Emotalk3d)	13.40	16.55	3.42
Ours	**6.74**	**7.63**	**1.44**

inaccuracies in AU prediction arising from label noise or training variations. For motion precision, we compared our motion decoder against existing methods such as Emotalk3d. As presented in **Tab** 2 , our approach achieves superior accuracy metrics.

7. More Results with visualization

Comparison results. Fig. 3 shows the extra qualitative comparison results evaluated on the Emotalk3D datasets using different comparing approaches respectively.

Emotion editing results. To further verify the robustness of EmoDiffTalk in diverse emotion and expression editing tasks, we conducted comprehensive tests on three typical cases from the EmoTalk3D dataset, covering basic emotion categories including sadness, happiness, anger, fear, confusion, worry, disgust, disappointment, anxiety, and shyness, as well as fine-grained expression movements such as smiling, mouth corners downturned, frowning, laughing, and eyebrow furrowing. The visualization results in Fig. 4, 5 and 6 intuitively demonstrate the ability of our method in emotion and expression editing. Faced with inputs of different identities and different emotional expressions, the model can maintain stable editing performance, reflecting excellent robustness.

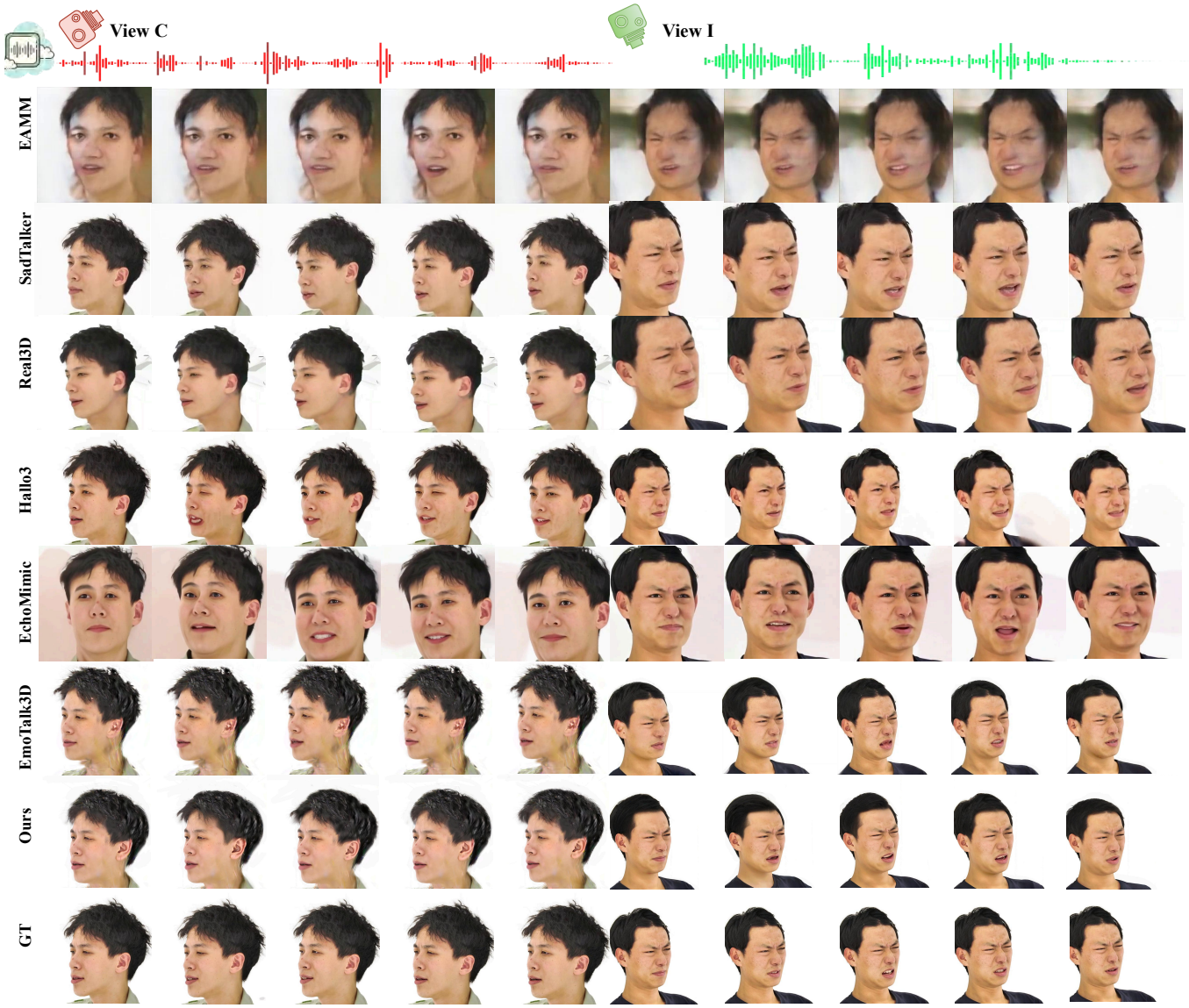

Figure 3. Extra qualitative comparison results evaluated on the Emotalk3D datasets using different comparing approaches respectively.

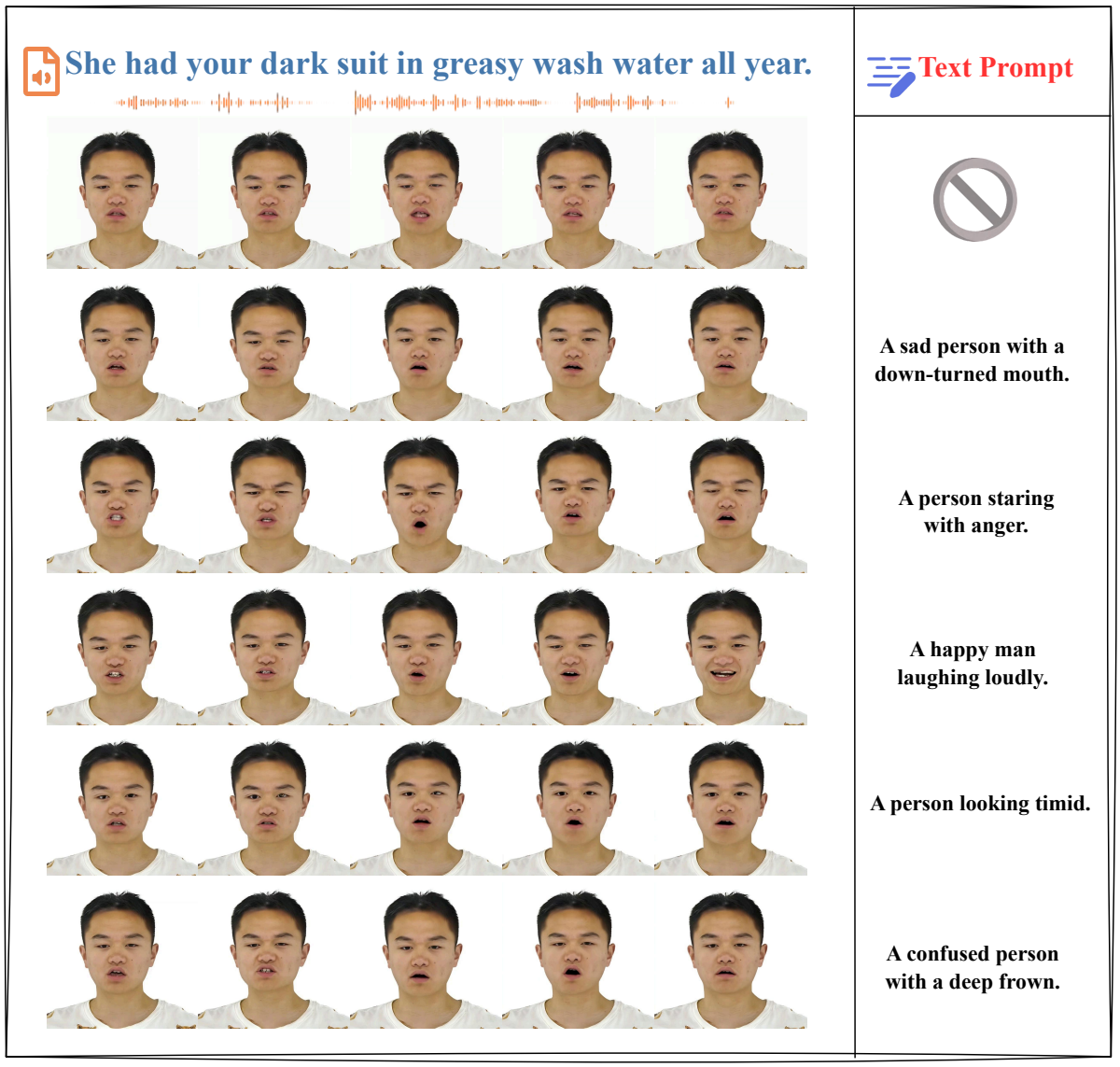

Figure 4. Visualization Results of Text-Guided Emotion and Expression Editing I

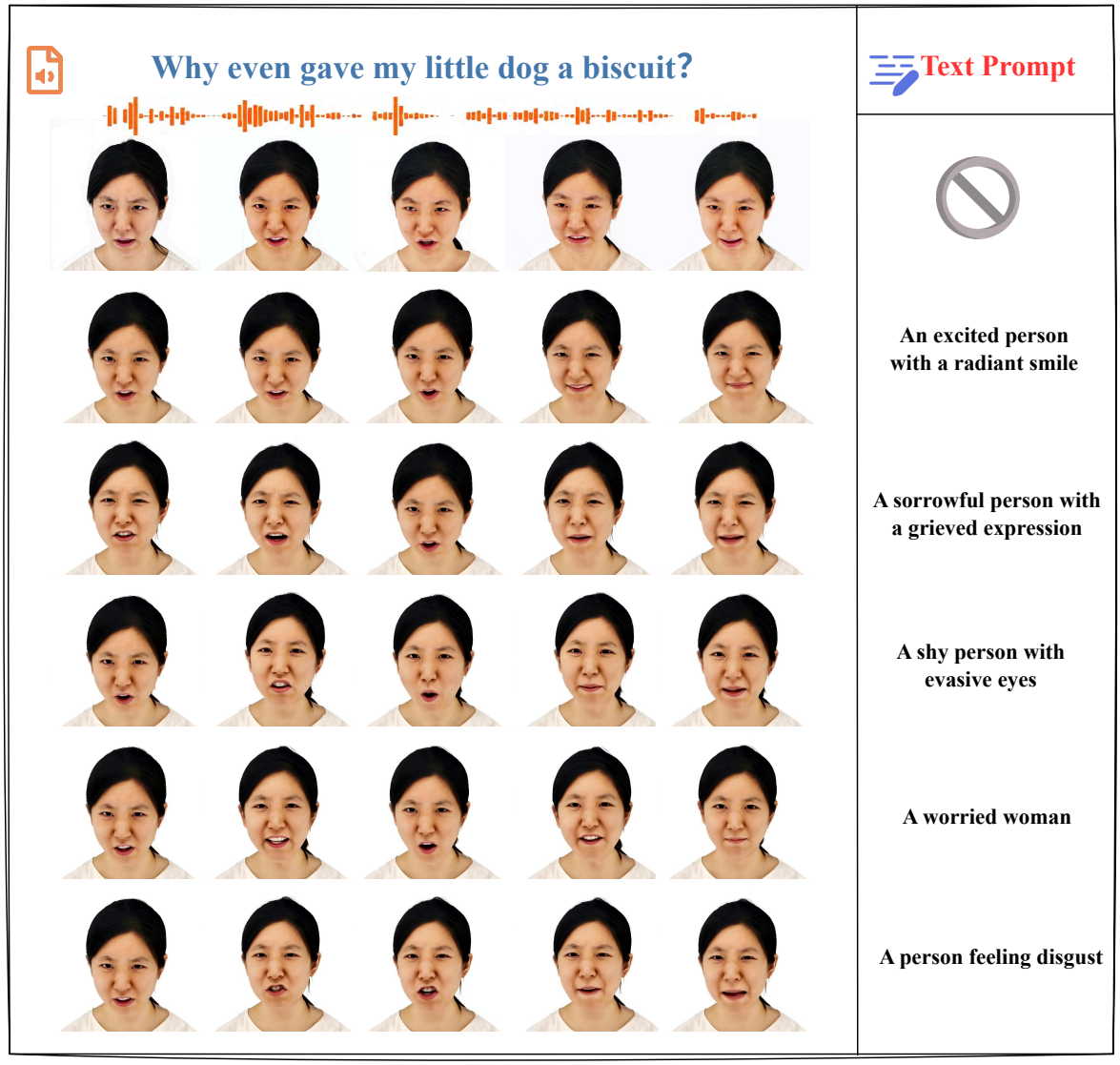

Figure 5. Visualization Results of Text-Guided Emotion and Expression Editing II

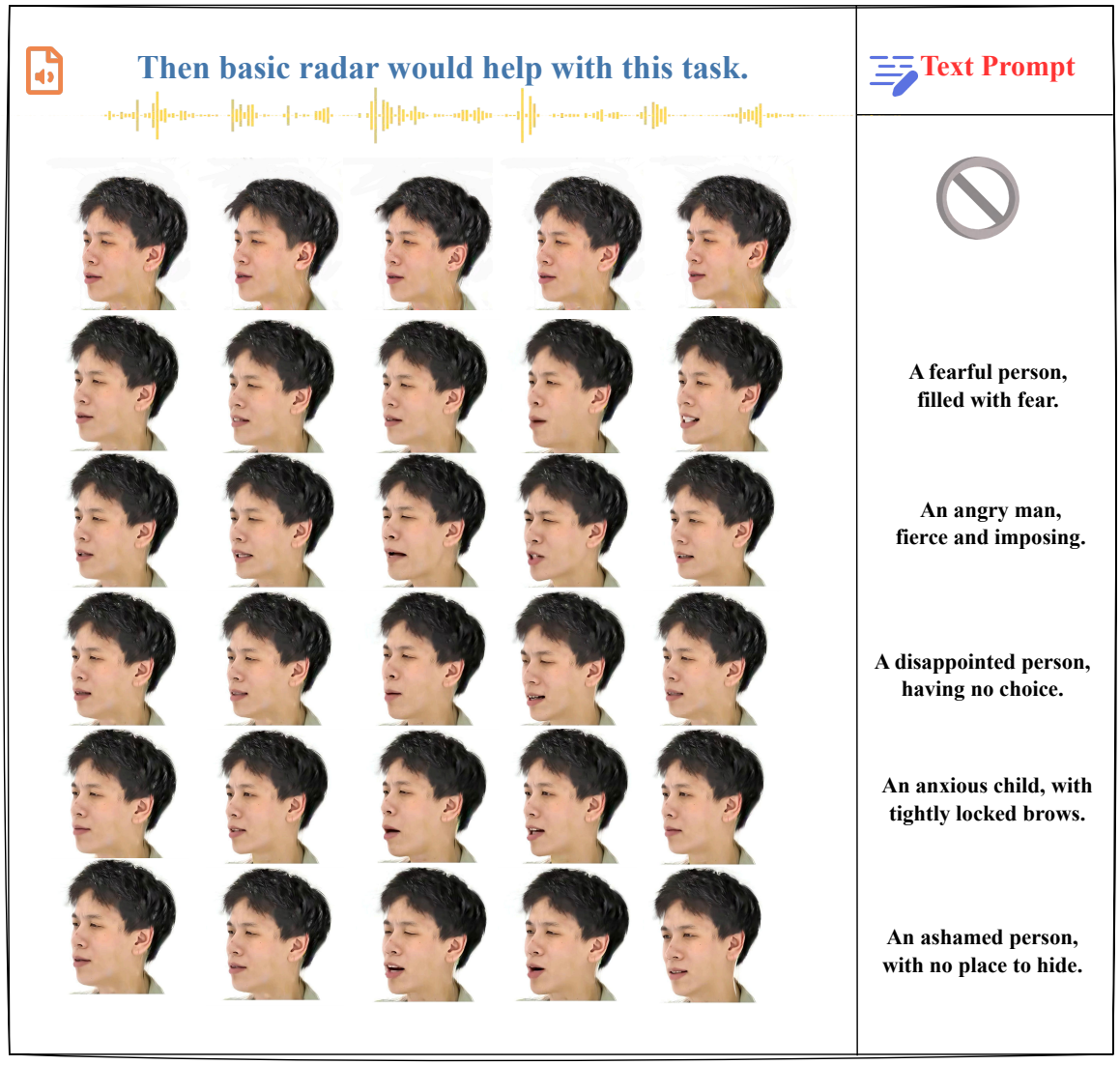

Figure 6. Visualization Results of Text-Guided Emotion and Expression Editing III

References

[1] Brandon Amos, Bartosz Ludwiczuk, Mahadev Satyanarayanan, et al. Openface: A general-purpose face recognition library with mobile applications. *CMU School of Computer Science*, 6(2):20, 2016. 2

[2] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, page 187–194, USA, 1999. ACM Press/Addison-Wesley Publishing Co. 2

[3] Qianyun He, Xinya Ji, Yicheng Gong, Yuanxun Lu, Zhengyu Diao, Linjia Huang, Yao Yao, Siyu Zhu, Zhan Ma, Songchen Xu, Xiaofei Wu, Zixiao Zhang, Xun Cao, and Hao Zhu. Emotalk3d: High-fidelity free-view synthesis of emotional 3d talking head. In *European Conference on Computer Vision (ECCV)*, 2024. 1

[4] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *CoRR*, abs/2106.07447, 2021. 1

[5] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1

[6] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 3

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 3

[8] Yating Wang, Xuan Wang, Ran Yi, Yanbo Fan, Jichen Hu, Jingcheng Zhu, and Lizhuang Ma. 3d gaussian head avatars with expressive dynamic appearances by compact tensorial representations, 2025. 2

[9] Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10324–10335, 2024. 1