When Words Smile: Generating Diverse Emotional Facial Expressions from Text

Anonymous ACL submission

Abstract

Enabling digital humans to express rich emotions has significant applications in dialogue systems, gaming, and other interactive scenarios. While recent advances in talking head synthesis have achieved impressive results in lip synchronization, they tend to overlook the rich and dynamic nature of facial expressions. To fill this critical gap, we introduce an end-to-end text-to-expression model that explicitly focuses on emotional dynamics. Our model learns expressive facial variations in a continuous latent space and generates expressions that are diverse, fluid, and emotionally coherent. To support this task, we introduce EmoAva, a large-scale and high-quality dataset containing 15,000 text–3D expression pairs. Extensive experiments on both existing datasets and EmoAva demonstrate that our method significantly outperforms baselines across multiple evaluation metrics, marking a significant advancement in the field. [1]

1 Introduction

In recent years, the remarkable success of dialogue systems has sparked a growing desire for face-to-face interaction with digital humans (Park et al., 2024). As emotional beings, humans rely heavily on facial expressions as a primary means of conveying emotions and intentions. Therefore, enabling digital humans to express emotions through facial expressions holds substantial research and application value (Sung-Bin et al., 2024).

A large portion of digital human (also referred to as talking head) research (Li et al., 2024; He et al., 2024), focuses primarily on the synchronization between speech and lip movements, while largely ignoring the rich emotional and expressive dynamics of face-to-face communication. Although some previous studies have recognized this limitation and started investigating potential solutions, most existing systems (He et al., 2024; Danecek et al., 2023; Pan et al., 2024) still generate only coarse facial expressions based on a limited set of discrete emotion labels (e.g., “happy,” “sad”), usually employing a pipeline architecture.

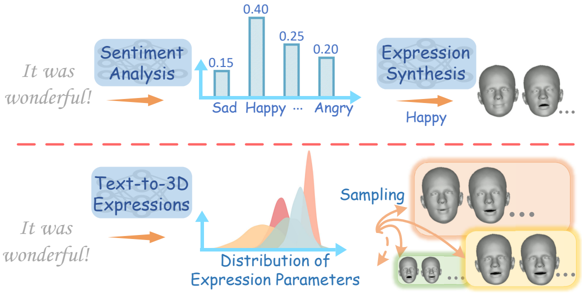

Figure 1: **Top:** The existing pipeline for synthesizing emotional avatars, which can only generate limited expressions that lack of diversity. **Bottom:** The proposed end-to-end system that directly maps text to facial expressions (codes), aims to generate diverse, emotionally consistent, and temporally smooth expressions.

Following the current pipeline, one typically performs sentiment analysis on the speech or text to obtain a discrete emotion label, which is then used to condition the facial expression generation (as shown in Figure 1). However, this approach faces at least two major limitations. First, emotion labels are typically limited in number and struggle to capture the full richness and subtlety of human emotional expression. Second, pipeline-based models are susceptible to information loss and error propagation across stages. End-to-end modeling could be one promising strategy to address the above limitations. By generating a continuous sequence of 3D facial expressions directly from the input utterance, the output is expected to be more diverse, natural, and emotionally consistent.

In this work, we present the first end-to-end work for text-3D facial expression learning. Technically, we propose a unified model, **CTEG** (Continuous Text-to-Expression Generator). CTEG leverages a

[1] Resources are available at (`https://anonymous.4open.science/r/EmoAva`).

CVAE-based autoregressive architecture to model expressive variations **in a continuous latent space**, enabling smooth and natural expression synthesis. To ensure emotional consistency between the input text and generated expressions, CTEG adopts a Latent Temporal Attention (LTA) mechanism that enhances the latent representation at each timestep by attending to historical context. Additionally, to promote expressive richness, CTEG incorporates an Expression-wise Attention (EwA) module that captures spatial dependencies among facial regions, enabling coordinated and varied facial movements.

To facilitate end-to-end training, we introduce **EmoAva**, a high-quality, large-scale dataset comprising 15,000 text-to-3D expression mapping instances, collected from multi-party dialogue scenes in professionally acted video sources. EmoAva provides rich, emotionally diverse, and context-aware expressive behaviors, offering a valuable foundation for studying facial expression generation in conversational AI. Extensive qualitative and quantitative experiments on the EmoAva as well as other representative existing datasets (Ng et al., 2023) demonstrate the superiority of our CTEG model in terms of expression diversity, naturalness, and emotional consistency, establishing a strong baseline for future research in text-to-expression generation.

In summary, our key contributions are as below.

- We propose a novel end-to-end model, CTEG, which learns text-to-expression mapping in a continuous latent space.
- We introduce **EmoAva**, a high-quality dataset with 15,000 annotated instances, designed to alleviate data scarcity in this domain.
- Extensive experiments demonstrate the effectiveness of CTEG in capturing expression diversity, naturalness, and emotional consistency, establishing a strong baseline for future research in this field.

2 Related work

Speech-driven Emotional Avatar Synthesis. Extensive research has been conducted on the synthesis of 3D talking heads (Richard et al., 2021; Ji et al., 2021; Papantoniou et al., 2022; Fan et al., 2022; Chu et al., 2018; Zhang et al., 2023b; Liu et al., 2024), most of which are speech-driven—generating lip-synced facial animations from audio input. These works generally overlook the modeling of facial expressions.

Recently, some approaches have started to integrate emotional context into the generation process (He et al., 2024). EMOTE (Danecek et al., 2023) addresses this by controlling expressions with single emotional labels, but the limited categories do not capture the full range of human emotions. Conversely, EmoTalk (Peng et al., 2023) and LaughTalk (Sung-Bin et al., 2024) extract tonal emotional features from speech to guide avatar synthesis similarly to talking head tasks. Complementary to these approaches, our method explores text as the sole source of emotional input since textual dialogue inherently conveys rich affective information and is more abundantly available than other modalities (Narayanan et al., 2009).

Text-Driven Human Motion Generation. Text-based human motion generation has significant applications in areas such as gaming and virtual reality. Much of the existing research in this domain focuses on generating sequences of human body movements (Zhang et al., 2023a; Jiang et al., 2023). In contrast, relatively little attention has been paid to text-driven human facial expression generation (Ng et al., 2023; Jung and Kim, 2025).

Our approach is most closely related to the recent work LM-Listener (Ng et al., 2023), which focuses on generating listener motions. In contrast, we concentrate on the more diverse and complex expressions of speakers. While LM-Listener employs a VQ-VAE-based framework, the use of discrete modeling and a constrained latent space may limit its ability to maintain temporal coherence and capture the full range of speaker-driven facial dynamics. In comparison, our method adopts a CVAE framework, whose structured continuous latent space is better suited for modeling the fluidity and expressiveness of facial behaviors.

3 Our Approach

3.1 Task Definition

Given a text input $\mathbf{x}$, our system generates a sequence of expression vectors $\psi = \{\mathbf{E}_0, \mathbf{E}_1, ..., \mathbf{E}_T\}$ over T time steps. The expression vectors are derived from 3D Morphable Face Model (3DMM) frameworks (Cao et al., 2014; Blanz and Vetter, 1999; Ng et al., 2023), which parameterize facial geometry in a compact and interpretable form. Among various 3DMMs, we follow Ng et al. (2023) and adopt the widely used FLAME model (Li et al., 2017), denoted as $\mathcal{F}$. Specifically, FLAME defines the parameter set

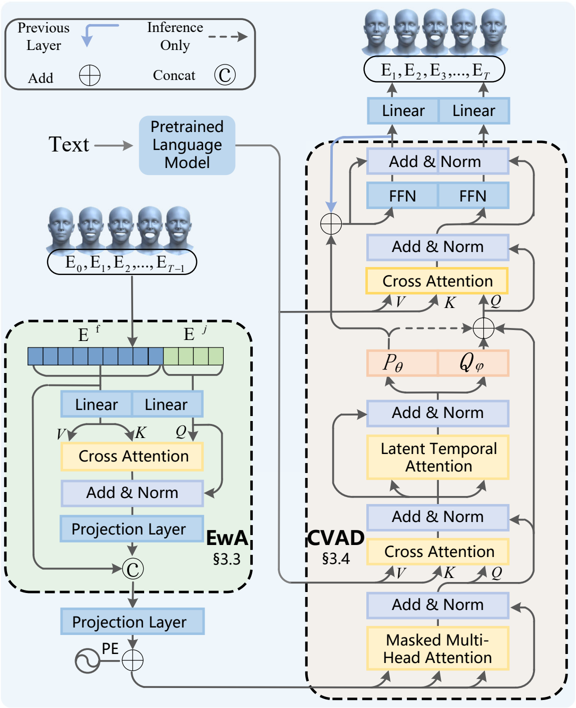

Figure 2: Architecture of the Continuous Text-to-Expression Generator (CTEG). Given a text, the model autoregressively generates a sequence of expression vectors. The green block and pink block represent the proposed Expression-wise Attention (EwA) module and the core Conditional Variational Autoregressive Decoder (CVAD) module, respectively.

$\mathcal{F} = \{\beta, \Delta v, \varrho, \Theta, \psi\}$, where β denotes shape, Δv represents vertex offsets, ϱ is global translation, Θ denotes joint poses, and ψ captures expression-related deformations. FLAME decouples expression from the shape and other identity-related factors, allowing us to directly regress the expression parameters $\psi \in \mathbb{R}^d$ in an identity-agnostic manner, where d (53 in this paper) denotes the dimension of the expression space.

3.2 Overall framework

The overview framework of the CTEG model is shown in Figure 2. CTEG primarily consists of the Expression-wise Attention (EwA) block at the encode side, and the Conditional Variational Autoregressive Decoder (CVAD) block at the decode side. From the perspective of architectural design, the EwA module serves as a feature enhancement module, while the CVAD functions as a hybrid of a CVAE (Sohn et al., 2015) and a transformer decoder (Vaswani et al., 2017). Technically, we adopt such an architecture for the following advantages. **1)** CVAE is beneficial for maintaining a smooth spatial distribution due to its nature of modeling in continuous space (Kingma and Welling, 2014; Sohn et al., 2015), which may help to model expression fluidity. **2)** Transformer decoder excels at modeling the long-range dependencies between sequences (Vaswani et al., 2017), which may help to model the emotion-content consistency. **3)** Due to the richness of the facial expression sequence, even within a single time step, facial expressions have countless variations. The Variational Autoregressive Decoder (VAD) may facilitate the modeling of diverse, time-varying sequences (Du et al., 2018). Given a text $\mathbf{x}$ as input, CTEG generates a sequence of expression vectors ψ autoregressively.

3.3 Expression-wise Attention Module

In the input part, we introduce EwA to establish connections between facial units and enhance the richness of the input expression in the feature space. This guides the subsequent CVAD module to capture different and rich patterns and structures, thereby improving the overall diversity of the model's generation results.

The expression vector $\mathbf{E}$ is constructed by concatenating two components: the jaw part $\mathbf{E}^j$ and the above-jaw part $\mathbf{E}^f$. Intuitively, these two parts are not independent of each other because human facial units function as a whole. For example, when a person laughs heartily, the jaw controls the opening of the mouth. To establish a connection between them, we first use a projection layer to map the raw expression vector to a latent space. Then we let the transfered $\mathbf{E}^j$ as the query, and let the transfered $\mathbf{E}^f$ as the key and value, feeding them into a cross attention module (Vaswani et al., 2017). After that, we apply dimensionality reduction to the output of the attention module and obtain $\mathbf{E}^{j'} \in \mathbb{R}^{|\mathbf{E}^j|}$. The final recombined 3D expression codes $\mathbf{E}' \in \mathbb{R}^{|\mathbf{E}|}$ are represented by: $\mathbf{E}' = \text{Concat}(\mathbf{E}^f, \mathbf{E}^j + \mathbf{E}^{j'})$. Then we project it into a high dimension d_{model}. In order to capture the order of expression sequence, we add the Positional Embeddings (PE) to the output of the EwA module. Specifically, we adopt the sinusoidal version positional encodings introduced in Vaswani et al. (2017).

3.4 Conditional Variational Autoregressive Decoder

Given a text $\mathbf{x}$ as input, an expression sequence ψ as output, CVAE is to maximize the conditional log-likelihood $\log p(\psi|\mathbf{x})$. To better capture temporal dynamics, we model the conditional probability distribution at each time step. Formally, the log-likelihood in our method is $\log \prod_{t=1}^{T} p(\psi_t \mid \psi_{<t}, \mathbf{x})$, rather than $\log p(\psi_{0,\dots,T} \mid \mathbf{x})$. To enhance the emotion-content consistency, we explicitly model the historical states of the latent variables. The resulting generation model can be formulated as:

$$\begin{aligned} p(\psi \mid \mathbf{z}, \mathbf{x}) &= \prod_{t=1}^{T} p(\psi_t \mid \psi_{<t}, \mathbf{z}_t, \mathbf{x}) \\ &= \prod_{t=1}^{T} p(\psi_t \mid \psi_{<t}, f_\zeta(\mathbf{z}_{<t}), \mathbf{x}) \,, \end{aligned} \tag{1}$$

where f_ζ is the Latent Temporal Attention (LTA) module, implemented by the masked multi-head attention (Vaswani et al., 2017). [2].

Intuitively, we assume the prior distribution P_θ and conditional distribution Q_ϕ to be multivariate Gaussian distributions:

$$\begin{aligned} Q_\phi(\mathbf{z}_t \mid \psi_{\le t}, \mathbf{x}) &= \mathcal{N}(\boldsymbol{\mu}_r(\psi_{\le t}, \mathbf{x}), \boldsymbol{\sigma}_r(\psi_{\le t}, \mathbf{x})) \,, \\ P_\theta(\mathbf{z}_t \mid \psi_{<t}, \mathbf{x}) &= \mathcal{N}(\boldsymbol{\mu}_p(\psi_{<t}, \mathbf{x}), \boldsymbol{\sigma}_p(\psi_{<t}, \mathbf{x})) \,. \end{aligned} \tag{2}$$

The two Gaussian distributions are parameterized by two neural networks respectively:

$$\begin{aligned} [\boldsymbol{\mu}_r, \boldsymbol{\sigma}_r] &= [h_r^{\mu}(\mathbf{o}), h_r^{\sigma}(\mathbf{o})] \,, \\ [\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p] &= [h_p^{\mu}(\mathbf{o}), h_p^{\sigma}(\mathbf{o})] \,, \\ \mathbf{o} &= \mathcal{A}_{mask}[\mathcal{A}(\psi_{\le t}, \mathbf{x}) \,, \end{aligned} \tag{3}$$

where h denotes a linear layer, $\mathcal{A}_{mask}$ and $\mathcal{A}$ denote masked attention module and cross attention module, respectively. As sampling $\mathbf{z}_t$ from two distributions is non-differentiable, we employ the reparameterization trick (Kingma and Welling, 2014):

$$\mathbf{z}_t = \mu_t + \sigma_t \odot \epsilon, \epsilon \sim \mathcal{N}(0, \mathrm{I}) \,, \tag{4}$$

$\mathbf{z}_t$ is drawn from $Q_\phi(\mathbf{z}_t \mid \psi_{\le t}, \mathbf{x})$ in the training stage, while drawn from $P_\theta(\mathbf{z}_t \mid \psi_{<t}, \mathbf{x})$ in the inference stage. After we obtain the sampled $\mathbf{z}_t$, we learn the second conditional generation distribution $P_\theta(\psi_t \mid \psi_{<t}, \mathbf{z}_t, \mathbf{x})$. Similarly, we assume the distribution a multivariate Gaussian distribution, the mean μ_g can be parameterized by the following generation network:

$$\begin{aligned} \mu_g^t &= \mathrm{FFN}(\mathrm{Concat}(\mathcal{A}(\mathbf{o}_1), \mathcal{A}(\mathbf{o}_2), \dots, \mathcal{A}(\mathbf{o}_l))) \,, \\ \mathcal{A}(\mathbf{o}_i) &= \mathcal{A}((\psi_{<t} + \mathbf{z}_{<t})W_i^Q, \mathbf{x}W_i^K, \mathbf{x}W_i^V) \,, \end{aligned} \tag{5}$$

where l is the number of cross attention heads. FFN is the position-wise feed-forward network.

Note that the CVAD module can be stacked multiple layers deep, where the input of the first layer comes from the EwA module, and the input of each subsequent layer comes from μ_g obtained by the previous layer. Specifically, the input at layer m is sampled from $\mathcal{N}(\mu_g^{m-1}, \sigma)$. For simplicity, we parameterize only μ_g and set the σ of the generative distribution to a matrix where all entries are equal to 1. Finally, we sample the predicted expression codes in time step t using the reparameterization trick again: $\hat{\psi}_t = \mu_g^{m,t} + \epsilon, \epsilon \sim \mathcal{N}(0, \mathrm{I})$. Our loss function of CVAD is as follows:

$$\mathcal{L}_{CVAD} = \sum_t \mathcal{L}_{rec}(\psi_t, \hat{\psi}_t) + \sum_t \mathcal{L}_{KL}(t) \,, \tag{6}$$

where $\mathcal{L}_{rec}$ is the mean squared error (MSE) loss, and the corresponding Kullback-Leibler (KL) divergence term is defined as follows:

$$\begin{aligned} \mathcal{L}_{KL}(t) = \mathrm{KL}\Big(& Q_\phi(\mathbf{z}_t \mid \psi_{\le t}, \mathbf{x}) \\ & \parallel \; P_\theta(\mathbf{z}_t \mid \psi_{<t}, \mathbf{x}) \Big) \,. \end{aligned} \tag{7}$$

3.5 Target Guided Loss

Despite the excellent performance, CVAD is known to suffer from a notorious issue referred to as *model collapse* (Fu et al., 2019; Yang et al., 2017). When this happens, the KL divergence term in the loss function becomes very close to zero, and the latent variable may be ignored by the decoder.

Some works have proposed various methods to mitigate this issue (Fu et al., 2019; Yang et al., 2017; Du et al., 2018). Among them, the most common approach is to adjust the weight of the KL term during the training process. However, this method requires carefully selecting parameters based on the specific training process of models, which is time-consuming when the dataset is very large or the model is very large. To this end, we design a simple yet effective loss function $\mathcal{L}_g$ to guide the latent variables to learn a meaningful structure. In this way, the latent variables may guide the autoregressive model in its generation process, thereby preventing the model from directly ignoring the latent variables. Formally,

$$\begin{aligned} \mathcal{L}_g &= \sum_t \mathcal{L}_{rec}(\psi_t, f_\gamma(\mathbf{o}_t)) \,, \\ \mathbf{o}_t &= \sum_{i=1}^{N_c} \mathrm{FFN}_i(\mathbf{z}_{<t}^i) \,, \end{aligned} \tag{8}$$

where f_γ is a linear projection layer, N_c is the number of the CVAD layers.

3.6 Details of Training and Inference

The total loss function of CTEG is:

$$\mathcal{L}_{total} = \mathcal{L}_{CVAD} + \mathcal{L}_g \,, \tag{9}$$

We freeze the pretrained language model parameters during training and update the remaining parameters using the backpropagation algorithm. We use the teacher forcing (Williams and Zipser, 1989)

[2] We also try another simple model, the details can be found in the Appendix C

Dataset	#Train	#Validation	#Test	Multi-person?
Ng et al. (2023)	2,366	222	543	✗
EmoAva	12,000	1,500	1,500	✓

Table 1: Comparisons of text-to-3D expression datasets. EmoAva is significantly larger and more diverse, featuring characters from over 100 screen productions.

approach to train in parallel for speed, but decode the expression codes sequentially during the inference phase. Adam (Kingma and Ba, 2015) optimizer is adopted here. We train 100 epochs with the maximum expression sequence length 256 and maximum sentence length 128. We set the warm-up steps *warmup* = 4000 and adopt the same learning rate scheduler in Vaswani et al. (2017):

$$lr = d_{model}^{-0.5} \cdot \min(step^{-0.5}, step \cdot warmup^{-1.5}) \, . \tag{10}$$

Additionally, we also employ residual connection (He et al., 2016) and layer normalization (Ba et al., 2016) throughout our architecture, as detailed in Figure 2. The number of heads in all our attention modules is set to 12. The inner hidden dimension for the FFN model is 2048. Following Ghosh et al. (2021), we adopt the pretrained language model BERT(Devlin et al., 2019) to obtain meaningful sentence embeddings. Specifically, the last hidden state of *bert-base-cased* is used here. Both the text embedding and the expression embedding share the same feature dimension $d_{model} = 768$. We only use one single CVAD layer in this paper (i.e., $N_c = 1$). More details about model analysis can be found in the Appendix C.

To ensure smoothness in the predicted sequences, an *average* expression—computed as the mean of all expression parameters in the training set—is prepended to each sequence. We treat an expression with all parameters set to zero as a terminator, referred to as the "*standard*" face. Since sampling in a continuous space differs from discrete space sampling in language models, optimizing a single point in continuous space as a terminator is more challenging. One straightforward approach involves setting a threshold (e.g., 1.0) during the inference phase, stopping when the predicted expression is very close to the "standard" face based on the Euclidean distance. We further employ a length-constrained decoding method by setting a maximum sequence length (MSL).

4 EmoAva Dataset

We expect each instance in EmoAva to include a piece of text to be spoken, and a corresponding sequence of 3D expression vectors, as illustrated in Figure 3. To construct this data, we first gather a large number of video clips from TV series and movies with dialogues. We consider two existing data sources for multimodal emotion analysis task, MELD (Poria et al., 2019) and MEMOR (Shen et al., 2020), both of which consist of television show segments. Besides these, we also gather numerous video clips from YouTube. A total of 21,390 such raw clips are collected, all in English.

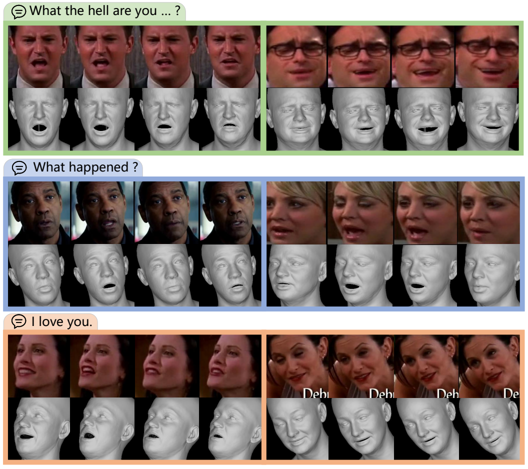

Figure 3: Samples from **EmoAva** dataset. Each instance includes a textual dialogue spoken by an actor, a corresponding head video, and a sequence of 3D expression vectors (here visualized in 3D mesh).

We apply various preprocessing methods. A brief overview is provided below, with details in the Appendix A. We employ WhisperX (Bain et al., 2023) to transcribe the audio, resulting in the corresponding text and timestamps. Afterwards, we cut the videos via the timestamps, creating dialogue video segments corresponding to the texts. To obtain clean headshot segments for each speaker, we develop a two-stage speaker localization pipeline. Specifically, we first apply FaceNet (Schroff et al., 2015) for automatic face tracking. To handle the challenges posed by complex visual scenes—such as multiple characters in a single frame or frequent speaker switches—we further perform manual refinement to ensure accurate speaker segmentation. After obtaining the head-videos, we adopt a 3D face tracking model EMOCA-v2 (Danecek et al., 2022) to extract the 3D expressions from 2D videos.

We collect a total of 15,000 text-to-expression pairs. A comparison with existing dataset is shown in Table 1. The dataset contains 782,471 FLAME frames. Among the 15,000 pairs, 2,270 exhibit a one-to-many (1-to-N) relationship—where N ranges from 2 to 76.

5 Experimental Setup

5.1 Evaluation Methods

As introduced in Section 1, our system aims to generate a sequence of expressions that are diverse, fluid, and consistent with the conveyed emotional content. To evaluate this, we adopt several evaluation metrics from prior studies and introduce additional fine-grained evaluation criteria. The calculation formulas for the following metrics are presented in the Appendix D.

Diversity measures the diversity of generated sequences without text as conditions (Zhang et al., 2023a). We randomly sample $N/2$ sequence pairs and calculate the average Euclidean distance between the expression vectors.

Multimodality (abbreviated as MModality) measures the diversity of text-conditioned results (Zhang et al., 2023a). We generate two expression sequences per text and compute their average Euclidean distance.

Variation measures the diversity of a sequence as it changes over time (Ng et al., 2023).

Fine-grained Diversity (abbreviated as FgD) quantifies the subtle temporal fluctuations within facial expression sequences that are not fully captured by existing diversity metrics like Variation (Ng et al., 2023). While Variation measures overall sequence diversity over time, FgD focuses specifically on the average Euclidean distance between adjacent frames to capture rapid, fine-grained changes in expressions.

Diversity on Test (abbreviated as DoT) measures the diversity of the expression sequences generated from the test set texts from a macro perspective.

Continuous perplexity (abbreviated as Cppl) evaluates how naturally an expression sequence evolves over time, reflecting the smoothness of the expression sequence and the uncertainty in a model's predictions. Contrary to the traditional discrete perplexity metric (Jelinek et al., 1977), Cppl is computed in continuous space.

Consistency assesses the extent to which the expressions accurately represent the emotions that would naturally correspond to a given utterance. Due to the absence of precise automatic tools to evaluate this alignment, we rely on human evaluation following Zhang et al. (2023a).

5.2 Experimental Settings

We mainly utilize the EmoAva dataset to validate the CTEG. We also conduct experiments on existing listeners' dataset (Ng et al., 2023) for reference in the Appendix C. The parameter settings of CTEG are detailed in §3.6. For each comparing method, we randomly sample 50 sentences from the test set and generate the corresponding expression sequence with a maximum length of 128. As mentioned in §5.1, there are currently no suitable quantitative metrics for emotion-content consistency evaluation. Instead, we adopt perceptual experiments following Guo et al. (2022). Each sample is rendered as an avatar video at 24 frames per second and shown to five participants. The participants are instructed to rank the outputs based on the emotional consistency between the facial expressions and the corresponding text.

5.3 Baselines

LM-Listener. To the best of our knowledge, this is the only open-sourced method applicable to the text-to-3D expression task (Ng et al., 2023). We implement the model with their released code, with most parameters kept unchanged. To ensure a fair comparison and obtain diverse outputs, we use *top-p* sampling (*top-p*=0.8) instead of greedy search.

Shuffle. Each expression sequence is randomly shuffled along the temporal axis to rigorously test the model's sensitivity to temporal coherence and expression fluency.

Random. Following Ng et al. (2023), we also randomly select expression sequences from the training set to assess the model's ability to model the emotion-content consistency.

6 Experimental Results

6.1 Main results

As shown in Table 2, CTEG outperforms the baselines across all diversity metrics by a large margin. We also visualize the diversity of expressions generated by CTEG in Figure 10 and 12 (Appendix). We randomly sample four sequences of expressions given a text. These examples demonstrate that the generated expressions exhibit a rich diversity.

Figure 4 illustrates an evaluation of user preferences. Participants are instructed to rank the expressions according to how well their emotions aligned with the input text. Compared with the *random* setting, LM-Listener, and CTEG, we find that the user preference for CTEG is higher than that of the

	Cppl ↓	DoT →	FgD →	Diversity ↑	MModality ↑	Variation →
GT	/	9.24	1.26	/	/	0.28
Shuffle	$9.90e6$	/	/	/	/	/
LM-Listener (Ng et al., 2023)	/	7.99	1.04 ± 0.0039	7.01 ± 0.0736	6.40 ± 0.0662	0.32 ± 0.0048
CTEG ($MSL = 256$)	**262.19**	**9.96**	**1.22** ± 0.0020	**8.18** ± 0.0508	**9.35** ± 0.0430	0.72 ± 0.0055
CTEG ($MSL = 64$)	/	8.91	1.22 ± 0.0021	7.75 ± 0.0535	8.48 ± 0.0485	**0.31** ± 0.0059

Table 2: Main Quantitative results. CTEG significantly outperforms the LM-Listener across four diversity metrics and achieves notably lower perplexity compared with the *Shuffle* setting. ↓ indicates that a lower value is better while ↑ suggests that a higher value is preferable. → indicates that the closer the value is to the *GT*, the better. / indicates not applicable. The standard error is estimated through bootstrap resampling with $1,000$ iterations.

	Cppl ↓	DoT →	FgD →	Diversity ↑	MModality ↑	Variation →
GT	/	9.24	1.26	/	/	0.28
w.o. EwA	**205.65**	6.97	1.14 ± 0.0018	6.07 ± 0.0246	6.55 ± 0.0293	0.40 ± 0.0025
w.o. LTA	243.57	8.60	1.20 ± 0.0030	7.58 ± 0.0484	8.01 ± 0.0671	0.58 ± 0.0062
w.o. $\mathcal{L}_g$	646.34	7.67	1.83 ± 0.0014	6.81 ± 0.0268	7.30 ± 0.0256	**0.38** ± 0.0017
CTEG	262.19	**9.96**	**1.22** ± 0.0020	**8.18** ± 0.0508	**9.35** ± 0.0430	0.72 ± 0.0055

Table 3: Quantitative results on the ablation study. CTEG model achieves the best overall performance compared with other settings. More in-depth experiments are provided in the Appendix C.

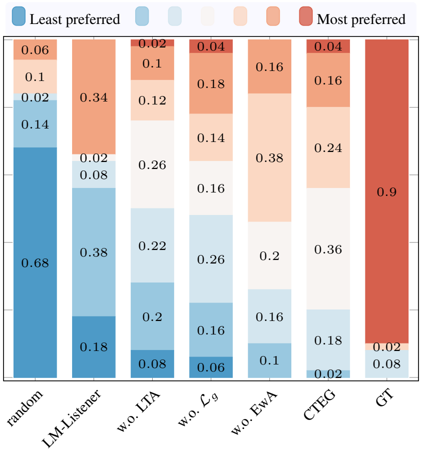

Figure 4: A quantitative evaluation of user preferences regarding emotion-content consistency. The color bar from blue to red indicates preference levels from lowest to highest. Expressions from CTEG better match text emotions than those from baselines.

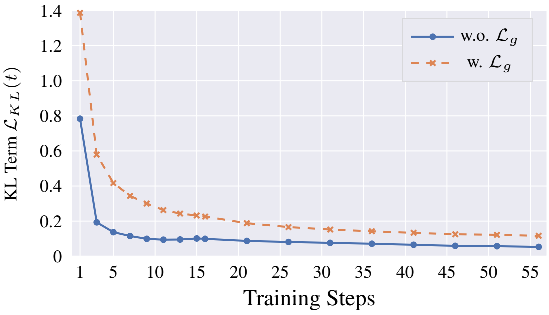

Figure 5: The effect of $\mathcal{L}_g$ loss (Eq. 8) on the KL term in Eq. 6. $\mathcal{L}_g$ loss mitigates the rapid decrease of the KL term and prevents it from approaching zero.

other two methods. This indicates the effectiveness of CTEG in modeling emotion-content consistency.

From Table 2, we observe that the *Cppl* metric for the *shuffle* setting is several orders of magnitude higher than that of the normal sequences, indicating CTEG's high sensitivity to the expression sequence order. The lower *Cppl* value confirms CTEG's effectiveness in modeling expression smoothness.

6.2 Ablation Study

We conduct a quantitative experiment on three key components (i.e., EwA module, LTA module and $\mathcal{L}_g$ loss function) in CTEG model. As shown in Table 3, removing the EwA module results in a significant drop in the four diversity metrics (*DoT*, *FgD*, *Diversity*, and *MModality*). This indicates that the EwA module makes a substantial contribution to the diversity of the generated sequences.

From Table 3 and Figure 4, we can observe that removing the LTA module results in a decrease in emotion-content consistency compared with the full CTEG model. This highlights the importance of the LTA module and supports the assumptions of our method. As shown in Table 3, after removing the $\mathcal{L}g$ loss function, only the *Variation* value shows improvement, while the performance of all other metrics declines. This indicates that removing the $\mathcal{L}g$ loss function leads to a drop in the overall performance of CTEG, and also reflects a weakened fitting ability of CTEG. These experimental results indicate that the $\mathcal{L}_g$ loss function effectively mitigates this issue, enhancing the model's generalization ability and overall performance.

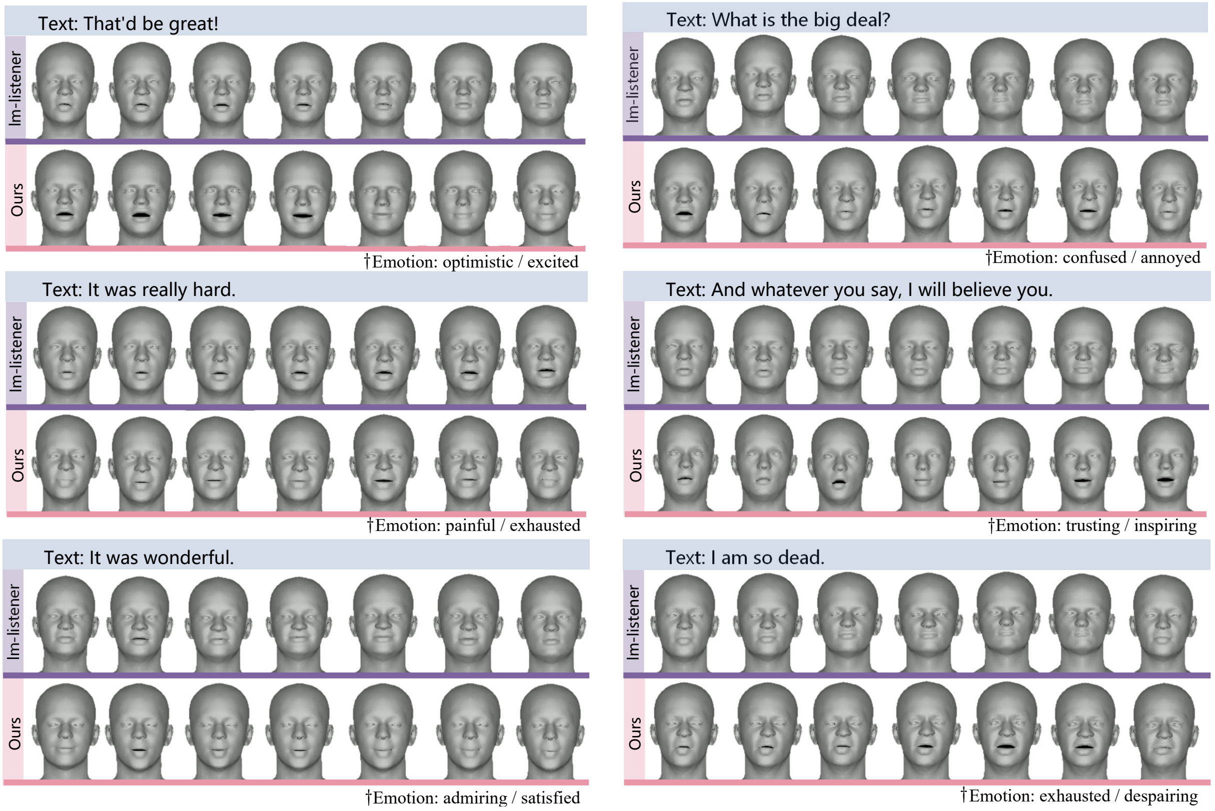
Text: That'd be great!
lm-listener
Ours
†Emotion: optimistic / excited
Text: What is the big deal?
lm-listener
Ours
†Emotion: confused / annoyed
Text: It was really hard.
lm-listener
Ours
†Emotion: painful / exhausted
Text: And whatever you say, I will believe you.
lm-listener
Ours
†Emotion: trusting / inspiring
Text: It was wonderful.
lm-listener
Ours
†Emotion: admiring / satisfied
Text: I am so dead.
lm-listener
Ours
†Emotion: exhausted / despairing

Figure 6: Qualitative analysis of the comparative results on emotion-content consistency. Our model demonstrates better consistency compared with the SoTA approach (LM-Listener). † represents the possible emotions conveyed by our results. More results are shown in the Figure 11.

6.3 Discussion

How does the $\mathcal{L}_g$ loss proposed in CTEG effectively mitigate the *model collapse* problem? As shown in Figure 5, we plot the changes in the KL term in the loss function (Eq. 9) as the training steps progress. It can be observed that after removing $\mathcal{L}g$, the KL term quickly drops and approaches zero, whereas with $\mathcal{L}g$, the KL term decreases more gradually, and the curve remains consistently above that of the *w.o.* $\mathcal{L}_g$ setting as the training steps increase. This phenomenon provides indirect support for the hypothesis proposed in §3.5.

Why does CTEG demonstrate stronger generative diversity in emotion? To understand why CTEG shows stronger diversity, we randomly sample from the latent space of CVAD and visualize the latent variable distribution (Figure 7). CTEG models more pattern clusters (146), a 29% increase compared to the *w.o.* *EwA* setting. This confirms that the EwA module enriches input features, resulting in a more diverse latent space in CVAD. The latent variables capture more varied patterns, which improves the diversity of generated outputs.

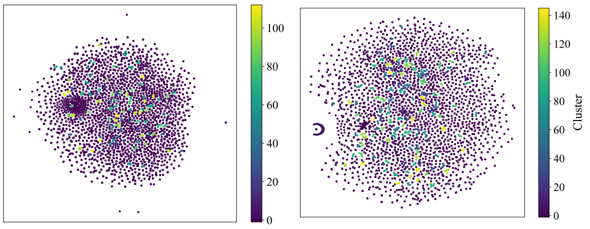

Figure 7: Comparison of latent variable distributions in the CVAD latent space. The *w.EwA* model (right) captures a broader range of generative modes (146 clusters) than the *w.o.EwA* model (113 clusters). This indicates that the EwA module enriches the feature space, enabling CVAD to model more diverse emotional patterns.

7 Conclusion

This paper proposes a novel end-to-end text-to-expression model, CTEG, which captures expressive variations in a continuous latent space, enabling the generation of diverse, fluid, and emotionally consistent facial expressions. To support this task, we construct EmoAva, a large-scale and high-quality dataset consisting of 15,000 instances. Extensive experiments demonstrate that CTEG significantly outperforms existing baselines across multiple aspects, paving the way for more emotionally aware digital humans.

Limitations

Limited language coverage. Currently, the EmoAva dataset is limited to English due to resource constraints. Building a comprehensive multilingual text–expression dataset is inherently challenging, but we view it as a promising future direction. In the Appendix, we detail our data collection pipeline, which we hope will serve as a foundation for future expansion and community collaboration. Although human languages are diverse, emotional expression is largely universal. This motivates us to extend our work toward multilingual emotional expression modeling in future iterations.

Limited personalization or identity adaptation. CTEG is identity-agnostic by design, aiming to model generalizable human emotional states across speakers. This design choice facilitates broader generalization but does not account for personalized expressive styles or speaker-specific traits. While personalized expression generation is an exciting future direction, we believe that building a strong foundation for modeling universal emotional patterns is a necessary first step—one that our work aims to establish.

Ethical Considerations

Annotator compensation. We employed three crowd-sourced annotators, all of whom are undergraduate students with strong English proficiency. They were compensated at an approximate rate of $10 per hour, which aligns with standard local compensation for similar tasks.

Copyright and privacy. We provide two licensing options for the dataset (detailed in A.1), defining the conditions under which it may be accessed and used. The original video materials used to construct the dataset are sourced from publicly available television series, and their copyrights remain with the respective rights holders. In addition, the facial features extracted for our model are identity-agnostic, meaning they do not retain personally identifiable characteristics of the actors. This serves as a form of de-identification, helping to preserve the portrait rights and privacy of the individuals appearing in the video content.

References

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. 2023. Whisperx: Time-accurate speech transcription of long-form audio. In *Proceedings of the INTERSPEECH*, pages 4489–4493.

Volker Blanz and Thomas Vetter. 1999. A morphable model for the synthesis of 3d faces. In *Proceedings of the SIGGRAPH*, pages 187–194.

Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. 2014. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425.

Hang Chu, Daiqing Li, and Sanja Fidler. 2018. A face-to-face neural conversation model. In *Proceedings of the CVPR*, pages 7113–7121.

Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. 2018. Voxceleb2: Deep speaker recognition. In *Proceedings of the INTERSPEECH*.

Darren Cosker, Eva Krumhuber, and Adrian Hilton. 2011. A facs valid 3d dynamic action unit database with applications to 3d dynamic morphable facial modeling. In *Proceedings of the ICCV*, pages 2296–2303.

Daniel Cudeiro, Timo Bolkart, Cassidy Laidlaw, Anurag Ranjan, and Michael J Black. 2019. Capture, learning, and synthesis of 3d speaking styles. In *Proceedings of the CVPR*, pages 10101–10111.

Radek Danecek, Michael J. Black, and Timo Bolkart. 2022. EMOCA: emotion driven monocular face capture and animation. In *Proceedings of the CVPR*, pages 20279–20290.

Radek Danecek, Kiran Chhatre, Shashank Tripathi, Yandong Wen, Michael J. Black, and Timo Bolkart. 2023. Emotional speech-driven animation with content-emotion disentanglement. In *Proceedings of the SIGGRAPH*, pages 41:1–41:13.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the NAACL-HLT*, pages 4171–4186.

Jiachen Du, Wenjie Li, Yulan He, Ruifeng Xu, Lidong Bing, and Xuan Wang. 2018. Variational autoregressive decoder for neural response generation. In *Proceedings of the EMNLP*, pages 3154–3163.

Yingruo Fan, Zhaojiang Lin, Jun Saito, Wenping Wang, and Taku Komura. 2022. Faceformer: Speech-driven 3d facial animation with transformers. In *Proceedings of the CVPR*, pages 18770–18780.

Joseph L Fleiss, Bruce Levin, and Myunghee Cho Paik. 2013. *Statistical methods for rates and proportions*. john wiley & sons.

Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. 2019. Cyclical annealing schedule: A simple approach to mitigating KL vanishing. In *Proceedings of the NAACL*, pages 240–250.

Anindita Ghosh, Noshaba Cheema, Cennet Oguz, Christian Theobalt, and Philipp Slusallek. 2021. Synthesis of compositional animations from textual descriptions. In *Proceedings of the ICCV*, pages 1396–1406.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. 2022. Generating diverse and natural 3d human motions from text. In *Proceedings of the CVPR*, pages 5152–5161.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the CVPR*, pages 770–778.

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. 2024. Emotalk3d: High-fidelity free-view synthesis of emotional 3d talking head. In *Proceedings of the ECCV*.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.

Xinya Ji, Hang Zhou, Kaisiyuan Wang, Wayne Wu, Chen Change Loy, Xun Cao, and Feng Xu. 2021. Audio-driven emotional video portraits. In *Proceedings of the CVPR*, pages 14080–14089.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. 2023. Motiongpt: Human motion as a foreign language. In *Proceedings of the NeurIPS*, pages 20067–20079.

Siyeol Jung and Taehwan Kim. 2025. Difflistener: Discrete diffusion model for listener generation. In *Proceeding of the ICASSP*, pages 1–5.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the ICLR*.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *Proceedings of the ICLR*.

Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. 2024. Talkinggaussian: Structure-persistent 3d talking head synthesis via gaussian splatting. In *Proceedings of the ECCV*.

Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. 2017. Learning a model of facial shape and expression from 4d scans. *ACM Transactions on Graphics*, 36(6):194:1–194:17.

Jin Liu, Xi Wang, Xiaomeng Fu, Yesheng Chai, Cai Yu, Jiao Dai, and Jizhong Han. 2024. Osm-net: One-to-many one-shot talking head generation with spontaneous head motions. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(8):6888–6900.

Ramanathan Narayanan, Bing Liu, and Alok Choudhary. 2009. Sentiment analysis of conditional sentences. In *Proceedings of the EMNLP*, pages 180–189.

Evonne Ng, Sanjay Subramanian, Dan Klein, Angjoo Kanazawa, Trevor Darrell, and Shiry Ginosar. 2023. Can language models learn to listen? In *Proceedings of the ICCV*, pages 10049–10059.

Ye Pan, Shuai Tan, Shengran Cheng, Qunfen Lin, Zijiao Zeng, and Kenny Mitchell. 2024. Expressive talking avatars. *IEEE Transactions on Visualization and Computer Graphics*, 30(5):2538–2548.

Foivos Paraperas Papantoniou, Panagiotis P Filntisis, Petros Maragos, and Anastasios Roussos. 2022. Neural emotion director: Speech-preserving semantic control of facial expressions in" in-the-wild" videos. In *Proceedings of the CVPR*, pages 18781–18790.

Se Park, Chae Kim, Hyeongseop Rha, Minsu Kim, Joanna Hong, Jeonghun Yeo, and Yong Ro. 2024. Let‘s go real talk: Spoken dialogue model for face-to-face conversation. In *Proceedings of the ACL*, pages 16334–16348.

Ziqiao Peng, Haoyu Wu, Zhenbo Song, Hao Xu, Xiangyu Zhu, Jun He, Hongyan Liu, and Zhaoxin Fan. 2023. Emotalk: Speech-driven emotional disentanglement for 3d face animation. In *Proceedings of the ICCV*, pages 20687–20697.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the ACL*, pages 527–536.

Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. 2018. Generating 3d faces using convolutional mesh autoencoders. In *Proceedings of the ECCV*, pages 704–720.

Alexander Richard, Michael Zollhöfer, Yandong Wen, Fernando De la Torre, and Yaser Sheikh. 2021. Meshtalk: 3d face animation from speech using cross-modality disentanglement. In *Proceedings of the ICCV*, pages 1173–1182.

Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Niessner. 2019. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the ICCV*.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the CVPR*, pages 815–823.

Sefik Ilkin Serengil and Alper Ozpinar. 2021. Hyperextended lightface: A facial attribute analysis framework. In *Proceedings of the ICEET*, pages 1–4.

Guangyao Shen, Xin Wang, Xuguang Duan, Hongzhi Li, and Wenwu Zhu. 2020. Memor: A dataset for multimodal emotion reasoning in videos. In *Proceedings of the ACM MM*, pages 493–502.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. In *Proceedings of the NeurIPS*, pages 3483–3491.

Kim Sung-Bin, Lee Hyun, Da Hye Hong, Suekyeong Nam, Janghoon Ju, and Tae-Hyun Oh. 2024. Laughtalk: Expressive 3d talking head generation with laughter. In *Proceedings of the WACV*, pages 6404–6413.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the NeurIPS*, pages 5998–6008.

Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. 2021. One-shot free-view neural talking-head synthesis for video conferencing. In *Proceedings of the CVPR*, pages 10039–10049.

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. 2017. Improved variational autoencoders for text modeling using dilated convolutions. In *Proceedings of the ICML*, pages 3881–3890.

Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew J Rosato. 2006. A 3d facial expression database for facial behavior research. In *Proceedings of the FG*, pages 211–216.

Jianhui Yu, Hao Zhu, Liming Jiang, Chen Change Loy, Weidong Cai, and Wayne Wu. 2023. Celebv-text: A large-scale facial text-video dataset. In *Proceedings of the CVPR*, pages 14805–14814.

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Yong Zhang, Hongwei Zhao, Hongtao Lu, Xi Shen, and Ying Shan. 2023a. Generating human motion from textual descriptions with discrete representations. In *Proceedings of the CVPR*, pages 14730–14740.

Wenxuan Zhang, Xiaodong Cun, Xuan Wang, Yong Zhang, Xi Shen, Yu Guo, Ying Shan, and Fei Wang. 2023b. Sadtalker: Learning realistic 3d motion coefficients for stylized audio-driven single image talking face animation. In *Proceedings of the CVPR*, pages 8652–8661.

Xing Zhang, Lijun Yin, Jeffrey F Cohn, Shaun Canavan, Michael Reale, Andy Horowitz, and Peng Liu. 2013. A high-resolution spontaneous 3d dynamic facial expression database. In *Proceedings of the FG*, pages 1–6.

Zheng Zhang, Jeff M Girard, Yue Wu, Xing Zhang, Peng Liu, Umur Ciftci, Shaun Canavan, Michael Reale, Andy Horowitz, Huiyuan Yang, and 1 others. 2016. Multimodal spontaneous emotion corpus for human behavior analysis. In *Proceedings of the CVPR*, pages 3438–3446.

In this **appendix**, we present the following parts. Section A: more details about the EmoAva dataset. Section B: supplementary related work about dataset. Section C: in-depth analysis about CTEG. Section D: details and formulas of some evaluation metrics. Section E: additional visualization results.

A More Details about EmoAva Dataset

Details of the dataset construction and license are provided here. All processing code and the dataset will be made available via the provided anonymous GitHub repository.

A.1 License

We will provide two separate licenses for the dataset: one for the video files and another for the 3D expression code. The latter is directly relevant to the task presented in this paper and will be released under the CC BY-NC 4.0 license. The video data, however, involves copyright considerations related to film and television content. Since such data can benefit a broader range of tasks beyond our primary focus, we plan to release it under a more restrictive license. We have consulted legal experts regarding the conditions for releasing this type of data. The full licensing documentation will be made available in our GitHub repository.

A.2 Construction Pipeline

The dataset construction pipeline is shown in Figure 8. Given a raw video that may contain multiple people and varying shots, our goal is to extract a single talking-face video with a fixed camera view, and then extract 3D coefficients.

To achieve the first step, we need to segment each raw video both spatially and temporally. Spatial segmentation involves isolating the talking face from multiple possibly co-occurring faces, while temporal segmentation involves extracting a continuous shot, typically where a person is speaking a complete sentence or segment.

Specifically, for a raw video, the audio is extracted and transcribed using speech recognition and ASR models (i.e., whisperX) (Bain et al., 2023), to obtain timestamps and textual content. These timestamps typically correspond to complete utterances by individual actors. Utilizing these timestamps, we segment the raw video temporally, effectively achieving time-based segmentation. Spatial segmentation, however, presents a more complex challenge.

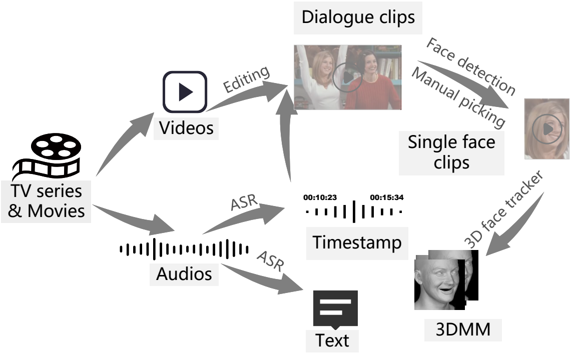

Figure 8: Pipeline for constructing the **EmoAva** dataset.

EmoAva	**Statistics**
# Train set	12,000
# Valid set	1,500
# Test set	1,500
# Dataset Total	15,000
Average sentence length	14
Average expression steps	52
Total frames of all expressions	782,471
# One-to-many instances	2270
# Expressions exceeding 256 frames	184

Table 4: Statistics of the EmoAva dataset.

To our best knowledge, no current method can reliably identify the speaking face among multiple faces in a frame. For the videos lacking speaker identity information, we exclude frames containing multiple detected faces. The construction algorithm involves multiple AI-based models, none of which can guarantee 100% accuracy. To ensure the quality of the dataset, we perform a manual check on the segmented videos. This process achieves our initial objective. After obtaining the segmented videos, we employ the widely-used FLAME tracking model EMOCA v2 (Danecek et al., 2022), to extract 3D coefficients. Following this, we conduct a manual check on the final instances to ensure their accuracy and quality.

In conclusion, we proposes a semi-automated approach that leverages several algorithms to generate large-scale instances. In this framework, human annotators are primarily tasked with verifying the algorithmic outputs and eliminating low-quality instances, thereby significantly enhancing efficiency and scalability.

A.3 Guidelines for Human Annotation

To maintain the data quality, we perform manual checking. Specifically, we employ three annotators to remove low-quality instances, where the criteria are as follows: **1)** The face should be clearly visible, without obstructions like masks or sunglasses.

2) The actor's facial expression changes should be continuous (i.e., no scene cuts). **3)** The actor should complete their sentence without being abruptly cut off. **4)** There should be only one person in the video from start to finish. **5)** The text should match what the actor is saying. **6)** The avatar expressions (mesh format driven by tracked vectors) align with those in the corresponding videos. We determine whether to drop data samples through independent annotation by the three annotators, followed by a majority vote on the results. After annotation, we calculate the Fleiss' kappa score (Fleiss et al., 2013), achieving a value of $\mathbf{0.86}$. This indicates minimal disagreement among the annotators, reflecting the high quality of the dataset's annotations.

A.4 Criteria for Collecting the Raw Videos

Crucially, manual screening is necessary when selecting television show segments from the internet (i.e., YouTube). First, we need to avoid cartoons or fantasy genres that do not feature real human faces. Second, we must steer clear of videos that may contain violence, gore, or explicit content that is not appropriate for mainstream audiences.

A.5 Dataset Insights

We randomly partition all instances in the training set into three subsets: training/validation/test sets, comprising $80\%/10\%/10\%$ of the total, respectively. We provide a brief summary of the key characteristics of the EmoAva dataset in the following and in Table 4.

Large-scale and High-quality. To ensure data quality, we employ SoTA methods at every stage of the dataset preprocessing algorithm. Additionally, we manually check and remove the expressions that lack fluidity or do not consistently match the emotions expressed in the text. As a result, our dataset comprises $15,000$ text-3D expression instances and a total of $782,471$ FLAME frames.

Diverse Mapping. In a dataset of $15,000$ text-expression pairs, there are $2,270$ instances with a 1-to-N relationship (where N ranges from 2 to 76), accounting for over 15%.

Diverse Expressions. EmoAva comes from a diverse range of sources and scenarios, i.e., including over 100 movies and more than 5 TV shows, thus resulting in a wide variety of expressions.

Diverse Emotion. Expressions exhibit a good diversity of emotions, including *Happy*, *Sad*, *Neutral*, *Disgust*, *Fear*, *Surprise*, and *Angry*.[3]

Rich and Varied Emotions. The emotions within a single sequence are also quite diverse, exhibiting significant variability. The proportion of expressions containing two or more types of emotions exceeds 95%.

Highly Scalable. The dataset includes raw videos, raw audio files, and a semi-automated construction algorithm, facilitating the extension to additional modalities and tasks.

B Supplementary Related Work

3D Avatar Head Dataset. This work also closely relates to the development and use of 3D Avatar Head Datasets. Numerous benchmark datasets (Yin et al., 2006; Cudeiro et al., 2019) exist within the community, but there is limited focus on benchmarks for emotion-aware dynamic 3D avatars. Current dynamic 3D avatar benchmarks typically lack language signals (e.g., text or speech) (Cosker et al., 2011; Ranjan et al., 2018; Zhang et al., 2013), a gap our research aims to fill.

To achieve Emotion-Content Consistency, an ideal 3D face dataset pairs each text instance with corresponding facial expression sequences as the text is articulated. Existing datasets (He et al., 2024; Zhang et al., 2016) often fall short, requiring annotators to read sentences with preset emotions, thereby ignoring the text's intrinsic emotional content. Notably, the MovieChat (Chu et al., 2018) dataset also includes 3D expression sequences. Unfortunately, they only release FACS muscle features, which can not reconstruct realistic facial details and significantly diverge from the current research framework.

To overcome this, we collect 2D video clips of actors' dialogues from various films and TV scenes, closely mirroring real conversational scenarios with consistent emotion-language alignment. We then use the SoTA FLAME tracking approach to extract 3D expression codes and meshes from 2D videos.

Talking Head Video Datasets. As introduced before, our goal is to collect videos of talking faces that exhibit rich, emotionally varied expressions in

[3] The statistical analysis is performed using a widely used emotion recognition framework DeepFace (Serengil and Ozpinar, 2021).

naturalistic conversational settings. Several existing datasets are relevant, including those from the talking head synthesis (Wang et al., 2021; Chung et al., 2018; Rossler et al., 2019), multimodal emotion analysis (Shen et al., 2020; Poria et al., 2019), and text-video modeling (Yu et al., 2023).

Among these, we find MELD (Poria et al., 2019) and MEMOR (Shen et al., 2020) to be the most suitable, as they are constructed from television show segments and contain real conversational dynamics with reasonably expressive faces. However, both datasets have notable limitations: they include only a small number of speakers, and their overall data scale is limited, which restricts their usefulness for training expressive generation models that aim for generalization and diversity.

In contrast, other commonly used datasets present further issues. For example, many talking head synthesis datasets involve individuals speaking directly to the camera with monotonous, emotionally flat expressions, lacking the nuanced variations seen in real social interaction. Additionally, several recent datasets collect videos from short-form content platforms (e.g., TikTok), but such self-recorded clips are often inconsistent in both expressive quality and visual fidelity, making them suboptimal for fine-grained expression modeling.

Based on this analysis, we find that movies and TV shows offer the most suitable source material, as they combine diverse emotional content, high production quality, and natural dialogue. To address the limitations in scale and speaker diversity of MELD and MEMOR, we construct a new dataset, further augmenting it with high-quality conversational clips sourced from YouTube. This hybrid strategy ensures a more scalable and emotionally rich dataset for expressive facial behaviors.

C In-depth Analysis of CTEG

In this section, we present more experimental results about the analysis of CTEG. Specifically, we investigate the following aspects:

Explorations on CTEG

- ★ **Q1**: How does CTEG perform when evaluated on the LM-Listener dataset?
- ★ **Q2**: Should the projection layer be shared between the input and the output of the CVAD module?
- ★ **Q3**: Can the attention mechanism in the LTA module be replaced with a simpler average pooling operation?
- ★ **Q4**: Should the weights of the pretrained language model (i.e., BERT) be fixed?
- ★ **Q5**: Does increasing the number of CVAD layers yield better performance?

	L2 ↓	FD ↓	Variation →	P-FD ↓
GT	/	/	0.11	/
LM-Listener	0.43 ± 0.02	18.22 ± 0.70	0.116 ± 0.005	19.63 ± 0.80
CTEG	**0.37** ± 0.03	**16.92** ± 1.20	**0.114** ± 0.007	**16.55** ± 0.90

Table 5: Quantitative results on the LM-Listener dataset.

C.1 More Details and Experimental Settings

We make several variants of CTEG in this paper. *w. sharing* refers to the model that shares the projection layers between the input and output of the CVAD module. Compared with it, CTEG does not share the projection layers. Based on *w. sharing*, we also test the performance of two variants, *w. pooling* and *w. BERT fine-tuning*. Compared to *w. sharing*, *w. pooling* modifies the LTA module by replacing the attention operation with an average pooling operation (i.e., averaging the latent variables from time steps 1 to s for time step s). Compared to *w. sharing*, *w. BERT fine-tuning* involves fine-tuning the BERT model during the training process, while the former fixes the weights of BERT. In addition to these, we also provide the performance of the ground truth (GT) on certain metrics for reference.

Our system is lightweight, with a total parameter count under 130M. All experiments in this paper are conducted on a single NVIDIA A100 GPU.

C.2 Results and Analysis

► Q1: How does CTEG perform when evaluated on the LM-Listener dataset?

We conduct a comparative experiment on the dataset used in Ng et al. (2023), following exactly the same evaluation settings. The results are presented in Table 5. As shown, CTEG consistently outperforms the LM-Listener model across all metrics, demonstrating its strong regression capability in the text-to-expression generation task.

Notably, we do not apply our proposed evaluation metrics to their dataset due to a fundamental mismatch in data characteristics. Specifically, their dataset lacks one-to-many mappings and includes only a single character. As a result, it does not account for emotional diversity—each input text corresponds to only one facial expression.

► Q2: Should the projection layer be shared

	Cppl ↓	DoT →	FgD →	Diversity ↑	MModality ↑	Variation →
GT	/	9.24	1.26	/	/	0.28
w. sharing	**241.28**	**8.53**	**1.27** ± 0.0032	7.16 ± 0.0436	8.01 ± 0.0292	0.53 ± 0.0027
w. pooling	300.10	7.89	1.30 ± 0.0102	3.17 ± 0.0225	6.47 ± 0.0605	**0.32** ± 0.0052
w. BERT fine-tuning	249.14	6.28	1.10 ± 0.0022	6.29 ± 0.0552	6.28 ± 0.0399	0.60 ± 0.0063
CTEG	262.19	9.96	1.22 ± 0.0020	**8.18** ± 0.0508	**9.35** ± 0.0430	0.72 ± 0.0055

Table 6: Quantitative results on some variants of CTEG. Lower values (↓) and higher values (↑) are preferred, while values closer to the Ground Truth (*GT*) are indicated by →. The standard error is estimated through bootstrap resampling with 1000 iterations.

between the input and the output of the CVAD module?

As shown in Table 6, *w. sharing* achieves the best results on the Cppl, DoT, and FgD metrics. The *Variation* also outperforms *CTEG* (w.o.sharing), but the *Diversity* and *MModality* metrics are lower than those of *CTEG*. Upon closer inspection, we find that the improvements in the Cppl and DoT metrics under the *w. sharing* setting are minimal. The difference between *CTEG* and *GT* for the *DoT* metric is $0.72(9.96 - 9.24)$, while the difference between *w. sharing* and *GT* is $0.71(9.24 - 8.53)$. One possible explanation for this phenomenon is the constraint on the generated vector space introduced by sharing the input and output mapping layers, which reduces diversity and yields metrics similar to those of *GT*.

Although *w. sharing* closely approximates GT on some metrics, we still remove the sharing operation in CTEG. The reason is that GT metrics serve only as reference values, and there are inherent differences between the test set used to compute GT and real-world data. Therefore, we can not use GT metrics (i.e., DoT, FgD, and Variation) as the sole criterion for evaluating the quality of our methods. In contrast, the performance of *w. sharing* on Diversity and MModality metrics is significantly inferior to that of CTEG. Considering these trade-offs, we believe that the sharing operation should not be retained in CTEG.

► Q3: Can the attention mechanism in the LTA module be replaced with a simpler average pooling operation?

When comparing *w. pooling* and *w. sharing* in Table 6, we observe a downward trend in many diversity metrics. This indicates that the attention operation in the LTA module is more effective than the pooling operation, which aligns with our intuition. For each current time step, a more reasonable integration of historical attention is beneficial, while simply averaging historical states may diminish the meaningfulness of the feature representation at the current moment, thereby weakening its rich representational capacity.

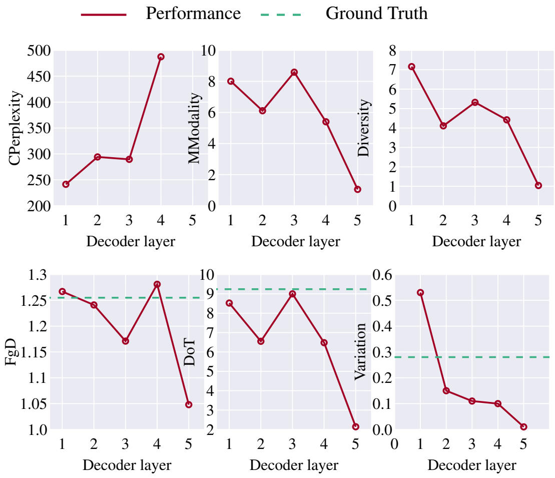

Figure 9: The performance of all three metrics shows a downward trend with the increasing number of decoder layers.

► Q4: Should the weights of the pretrained language model (i.e., BERT) be fixed?

In the comparison between *w. BERT fine-tuning* and *w. sharing*, almost all metrics showed a decline in performance in Table 6. We think this may be due to the limited size of the training set, which could have led to overfitting when fine-tuning the text embedding model.

► Q5: Does increasing the number of CVAD layers yield better performance?

We set the number of decoder layers from 1 to 5 to observe the diversity and naturalness of the generated expressions. As shown in Figure 9, it is evident that as the number of decoder layers increases, the model's performance gradually declines. Particularly when the number of layers reaches 5, the perplexity explodes, increasing by several orders of magnitude compared to the 4-layer decoder, and the diversity also becomes very poor. We find that the model struggles to converge under many-layer conditions. We speculate that this is because each step s in every layer is independently sampled, and N layers would generate s^N latent variable states, introducing too much randomness. We refer to this phenomenon as Cumulative Sampling Instability. Therefore, for the method described in this paper, using a single-layer decoder is the optimal configuration.

D Details of Evaluation Metrics

Diversity metric is calculated by the following formula (Zhang et al., 2023a):

$$\text{Diversity} = \frac{1}{N_d} \sum_{i=1}^{N_d} \left\| \Psi_i - \Psi_i^{'} \right\| , \qquad (11)$$

where Ψ and Ψ' denote a pair of randomly sampled sequences of expression vectors that are generated without giving any text. We set N_d to 750 in this paper.

MModality is calculated by the following formula (Zhang et al., 2023a):

$$\text{MModality} = \frac{1}{N_m} \sum_{i=1}^{N_m} \left\| \psi_i - \psi_i^{'} \right\| . \qquad (12)$$

We set N_m to 1500 in this paper. ψ_i and ψ_i' represent two different sequences generated under the same set of given texts.

Variation is calculated by the following formula (Ng et al., 2023):

$$\text{Variation} = \frac{1}{N_v} \sum_{i=1}^{N_v} \left(\frac{1}{n_i} \sum_{j=1}^{n_i} \text{var}(\text{E}_{ij}) \right) , \quad (13)$$

where E_{ij} denotes a frame in a sequence of expression vectors. n_i is the length of the i-th sequence. N_v here is the number of sequences, which is set to 1500. $\text{var}(\cdot)$ operation calculates the variance.

Fine-grained Diversity (FgD) is calculated by:

$$\text{FgD} = \frac{1}{(T-1)N} \sum_{i=1}^{N} \sum_{j=0}^{T-1} \left\| \mathbf{E}_{i,j+1} - \mathbf{E}_{i,j} \right\| . \qquad (14)$$

Diversity on Test (DoT) is obtained by calculating the average Euclidean distance between each pair of generated expression sequences:

$$\text{DoT} = \frac{2}{N(N-1)} \sum_{1 \le i < j \le N} \left\| \mathbf{E}_i - \mathbf{E}_j \right\| . \quad (15)$$

Continuous perplexity (Cppl). Given the i-th expression sequence ψ^i, we define the following entropy inspired by Jelinek et al. (1977).

$$H_i(\xi) \approx -\frac{1}{T} \sum_{j=1}^{T} \log_2 p_\xi(\psi_j^i \mid \psi_{<j}^i, \mathbf{x}) , \qquad (16)$$

p_ξ here is a continuous conditional distribution where modeled by a generation model. Multivariate normal distribution is adopted in this paper and it is calculated by:

$$\begin{aligned} p_\xi(\psi_j \mid \psi_{<j}, \mathbf{x}) \approx \Phi(x+\delta; \mu_\xi^j, \sigma^2\mathbf{I}) \\ - \Phi(x-\delta; \mu_\xi^j, \sigma^2\mathbf{I}) , \quad (17) \end{aligned}$$

where $\Phi(\cdot; \mu_\xi, \sigma^2\mathbf{I})$ denotes the cumulative distribution function (CDF) of the multivariate normal distribution with mean μ and covariance matrix $\sigma^2\mathbf{I}$. Note that δ and σ are empirical values, which are set to 0.8 and 0.2 here. Given N expression sequences, Cppl is calculated by:

$$Cppl = 2^{\frac{1}{N}\sum_{i=1}^{N} H_i(\xi)} . \qquad (18)$$

E More Visualization Results

In this section we present extended visualization results of the expressions generated by CTEG and the baseline method (Ng et al., 2023). As shown in Figure 11, we present several sequences of expressions generated by CTEG and LM-Listener (Ng et al., 2023). Compared with the LM-Listener, the expressions we generated exhibit a greater alignment with the emotions conveyed by the corresponding text. For example, the text “Oh damn, I picked the wrong side.” conveys emotions of pain, regret, and complaint. The expressions we generated effectively reflect these emotions. In contrast, the expressions produced by LM-Listener appear to convey a smile, which is inconsistent with the emotional tone of the text. A similar observation is evident in the text “What a beautiful story.” This statement conveys feelings of joy and admiration, and the expressions we generated reflect this joy. However, the expressions produced by LM-Listener appear to convey a sense of indifference. Many additional cases also support this observation in Figure 11.

Figure 10 and 12 present the visual expressions generated by CTEG with several different random seeds. Taking the text “I thought I was pretty good too.” as an example, this statement conveys emotions of pride, joy, or happiness. In the first sequence of expressions, the portrayal is one of delight. The second sequence exhibits a more subdued happiness, while the third sequence conveys a sense of pride and self-satisfaction. Despite the sig-

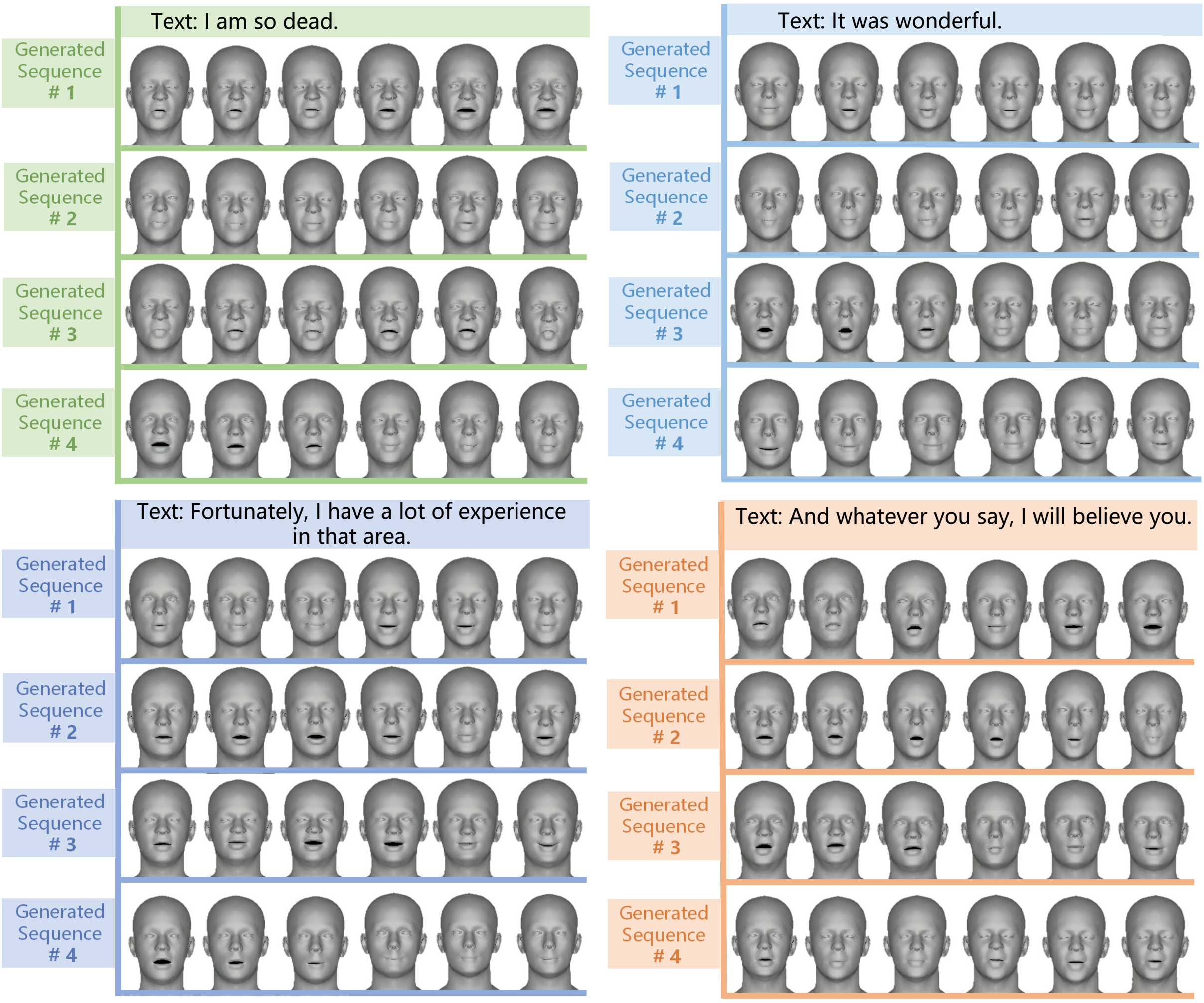

Figure 10: Visualization of the diversity generated by the CTEG model. Four sequences of expressions are generated from the same text with different random seeds. CTEG exhibits excellent generative diversity.

nificant differences among these three sequences of expressions, all align well with the emotions conveyed by the text. Similarly, taking the text “What the hell?” as an example, this statement generally conveys feelings of surprise, frustration, or disbelief. The three generated sequences of expressions all seem to convey these emotions. However, each sequence exhibits varying degrees of intensity in expressing surprise or frustration. In addition to these, this phenomenon can also be observed in other examples.

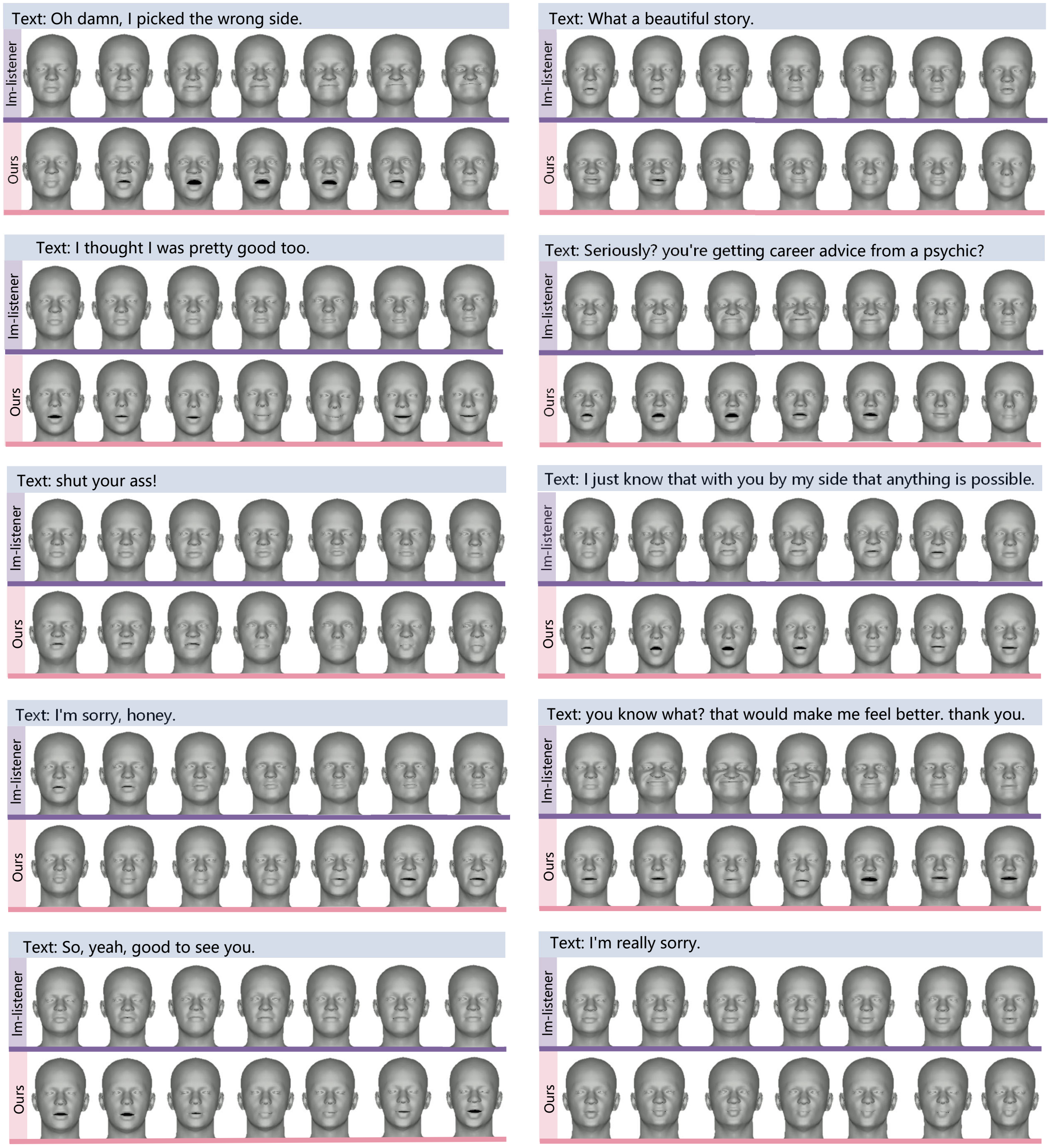
Text: Oh damn, I picked the wrong side.
lm-listener
Ours
Text: What a beautiful story.
lm-listener
Ours
Text: I thought I was pretty good too.
lm-listener
Ours
Text: Seriously? you're getting career advice from a psychic?
lm-listener
Ours
Text: shut your ass!
lm-listener
Ours
Text: I just know that with you by my side that anything is possible.
lm-listener
Ours
Text: I'm sorry, honey.
lm-listener
Ours
Text: you know what? that would make me feel better. thank you.
lm-listener
Ours
Text: So, yeah, good to see you.
lm-listener
Ours
Text: I'm really sorry.
lm-listener
Ours

Figure 11: Extended visual results generated by CTEG and LM-Listener (Ng et al., 2023). The expressions produced by CTEG exhibit greater consistency with the emotions conveyed by the corresponding text.

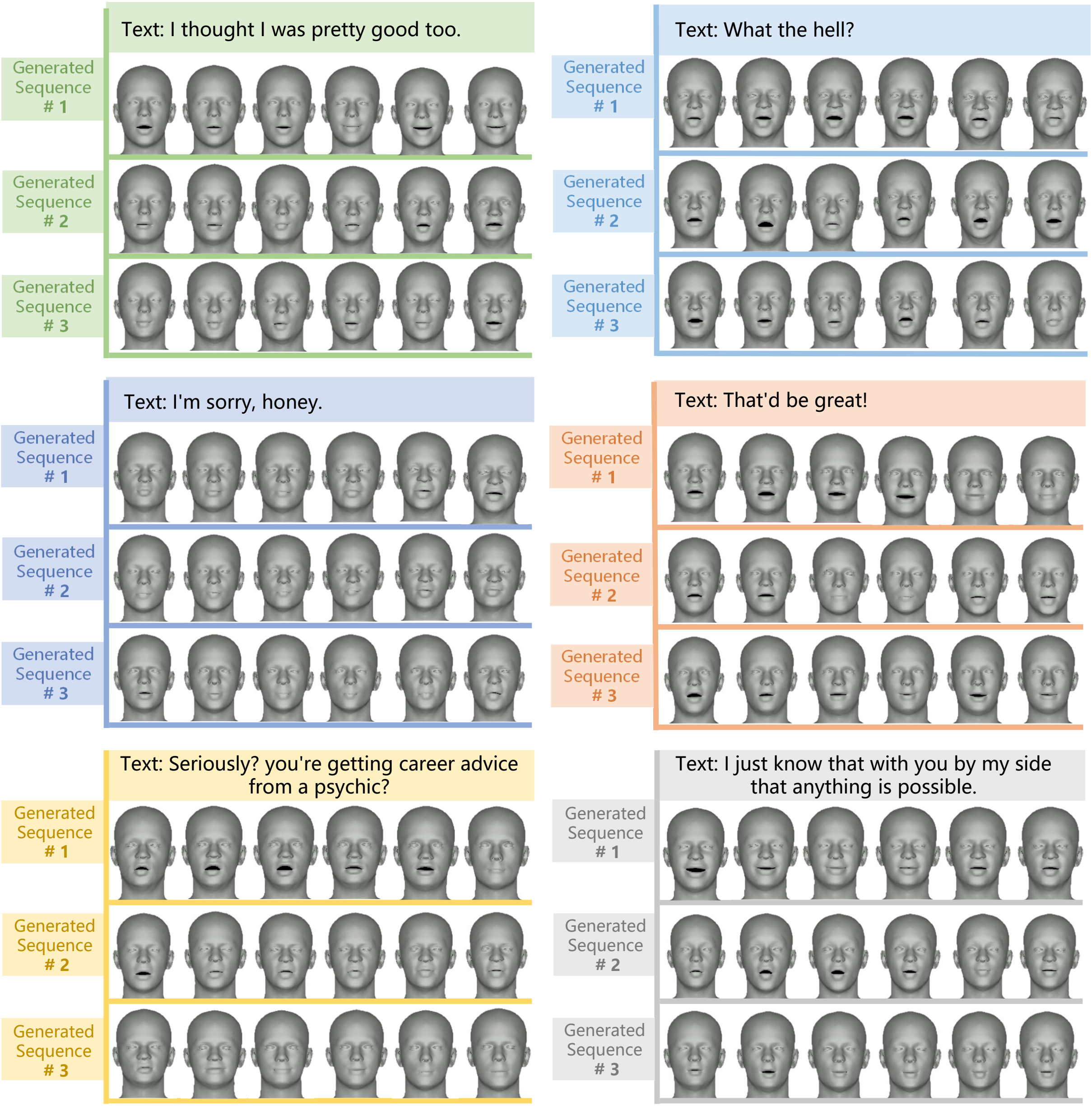

Figure 12: Visualization of the diversity generated by the CTEG model. Three sequences of expressions are generated from the same text with different random seeds. CTEG exhibits excellent generative diversity.