LEMO: LLM-GUIDED FINE-GRAINED EMOTION EDITING FOR SPEECH-DRIVEN FACIAL MOTION GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Although speech-driven facial motion generation has advanced, faithfully synthesizing expressions that match emotional prosody remains a challenging task. This is mainly due to the close coupling of emotional cues and phonetic content within audio signals, which results in learned representations where these aspects are entangled. From our analysis, we observe two prevalent failure modes in current models: *Expression Attenuation*, where facial animations lack the intended emotional intensity even when the audio input is expressive, and *Emotion Mismatch*, where the model reproduces emotions from phonetically similar training samples instead of the user's desired emotion. Rather than addressing these limitations solely during generation, we propose LEMO, a motion-level emotion editing framework. LEMO decomposes facial motion into upper-face movement, phonetically driven neutral lip motion, and expressive lip offsets, using a phone-driven decomposition approach. To support flexible, text-based editing, expressive offsets are discretized as structured region-state descriptors, achieved by spatial and temporal quantization of blendshape activation. This tokenized interface enables a large language model to interpret natural language instructions and deliver region- and time-specific expression changes. A diffusion-based temporal inpainter, together with a region-aware ControlNet, then applies these modifications additively, preserving lip synchronization. Quantitative experiments on the MEAD dataset show that LEMO sets a new benchmark for both expression fidelity and motion naturalness.

1 INTRODUCTION

Speech-driven 3D facial animation is foundational for digital avatars, virtual assistants, and content creation (Hu et al., 2024; Qian et al., 2024; Cudeiro et al., 2019). The ability to generate lifelike, temporally coherent, and acoustically synchronized facial motion from speech audio has opened new avenues across human-computer interaction. However, producing facial animations that convey the intended emotional expression remains a significant challenge.

Audio signals carry two fundamentally different types of information critical for realistic facial animation (Li et al., 2025; Peng et al., 2023). The first is **prosodic**, encompassing paralinguistic cues such as pitch contour, speaking rhythm, and intensity, which together convey emotion. The second is **phonetic**, consisting of linguistic content encoded in phones and phonemes, which primarily drives lip movements.

These two signals are inherently coupled, creating two critical failure modes in existing generation-based approaches: **Failure Mode 1: Expression Attenuation.** Even when the input audio carries clear emotional prosody, the generated animation often lacks intended emotional intensity, appearing flat or emotionally muted. **Failure Mode 2: Emotion Mismatch.** When the phonetic content closely resembles a training sample with a different emotional label, the model defaults to the training emotion. For instance, if "the news is good today" was delivered with excited prosody in training, a user requesting subdued tone for "the news is okay today" still receives an excited expression; forcing the subdued expression instead causes lip-sync drift. [1]

[1] We empirically demonstrate and analyze these failure modes in the Sec. 5.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Both failure modes share a common root cause: prosody and phonetic signals are deeply entangled in learned representations, making independent control structurally difficult. Existing approaches such as EMOTE (Daněček et al., 2023), MMHead (Wu et al., 2024), and ExpCLIP (Zhong et al., 2024) remain within the generation paradigm, applying coarse, sequence-level control without the ability to perform region-specific or temporally localized edits.

We argue that directly resolving the prosody-phonetic entanglement at the generation stage is structurally limited. Instead of fighting this entanglement, we propose to sidestep it entirely through a motion-level editing paradigm. The core insight is that if we first obtain a speech-driven animation with phonetically accurate lip movements, we can selectively edit expressive components as additive offsets without disturbing lip synchronization.

Specifically, we decompose facial animation into upper-area motion, phonetically driven neutral lip motion, and expressive lip offsets, then discretize these offsets into structured region-state descriptors via spatial and temporal quantization. These descriptors provide an interpretable interface that enables large language models (LLMs) to interpret free-form text instructions and generate precise region- and time-specific edits. We develop a diffusion-based temporal inpainter coupled with a region-aware ControlNet. The inpainter modifies expressive features, while the ControlNet injects context-aware edit signals to enforce fine-grained spatial and temporal constraints. Neutral lip motion remains unmasked to preserve lip sync. Overall, our contributions are three-fold:

- A prosody-phonetic aware motion editing framework that addresses the entanglement problem by editing pre-generated animation rather than controlling expression at generation, naturally preserving lip synchronization.
- A discretized offset representation that converts blendshape activations into structured region-state descriptors, enabling LLMs to serve as intelligent agents for fine-grained expression editing from free-form text.
- LEMO, a unified framework coupling LLM-agent reasoning with diffusion-based inpainting and region-aware ControlNet for precise, region-level and time-localized expressive editing.

2 Related Work

Audio-driven 3D Facial Animation has evolved from rule-based approaches (Edwards et al., 2016; Taylor et al., 2012) to data-driven techniques leveraging deep learning. Early methods such as FaceFormer (Fan et al., 2022), MeshTalk (Richard et al., 2021), and CodeTalker (Xing et al., 2023) established foundational pipelines for speech-to-animation synthesis. Recent work has focused on expressiveness: EmoTalk (Peng et al., 2023) decomposes speech content from emotional expressions, EMOTE (Daněček et al., 2023) introduces lip-reading consistency with emotion labels, and ExpCLIP (Zhong et al., 2024) extends control through textual descriptions. Diffusion-based methods like DiffPoseTalk (Sun et al., 2024), DEEPTalk (Kim et al., 2025), and 3DiFACE (Thambiraja et al., 2025) further capture dynamic emotion variations. However, these methods operate end-to-end where prosodic and phonetic signals remain coupled during synthesis, limiting independent control of expression and lip articulation. Our approach instead adopts a motion-level editing paradigm: we generate a phonetically accurate base animation and add expressive offsets to avoid prosody-phonetic entanglement.

Text-guided Facial Motion Generation has shown promise in 3D motion synthesis (Tevet et al., 2022a;b). For facial animation, MMHead (Wu et al., 2024) employs autoregressive frameworks connecting text with per-frame expressions, while ExpCLIP (Zhong et al., 2024) leverages text-to-expression CLIP models. However, these methods describe entire sequences with a single textual state, lacking fine-grained, region-specific or temporally localized control. Rather than conditioning directly on text, we use large language models to interpret instructions and reason about precise edits through our discretized region-state representation.

Emotional Expression Control. Existing methods remain constrained to coarse-grained control: EMOTE applies global labels to entire sequences, EmoTalk conditions on fixed emotion levels, and diffusion-based methods generate full animations from static specifications. We propose a fundamentally different strategy: rather than decomposing prosody and phonetic signals during generation, we avoid their entanglement by editing pre-generated animations. Our phone-driven method

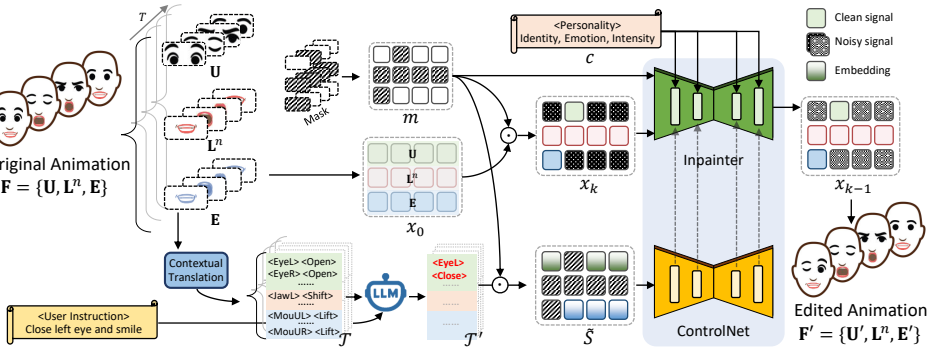


Figure 1: Our method comprises two core modules: an Inpainter for masked facial animation inpainting and a region-aware ControlNet for fine-grained expression editing. Given an input facial animation sequence $\mathbf{F}$, which is decomposed into upper-area motion $\mathbf{U}$, phone-driven neutral lip motion $\mathbf{L}^n$, and prosodic expression offset $\mathbf{E}$, along with an expressive mask m, specified personality conditions c, and string embeddings $\tilde{\mathcal{S}}$, LEMO enables the LLM-guided editing of expressive components in facial animation while preserving accurate and natural speech-driven lip movements.

separates facial motion into upper-area motion, neutral lip motion, and expressive lip offsets, enabling precise edits while preserving lip synchronization.

3 Data Preprocessing

This section explains how we prepare training and testing data by decomposing facial animation into phone-driven phonetic and prosodic components, forming the foundation for our editing framework.

3.1 Facial Animation Representation

We represent facial animation using ARKit Blendshapes (Apple Developer, 2025), which provide explicit semantic control over individual facial regions. We divide this facial structure into two primary areas: the upper (eyebrows, eyes, nose) and the lip (lips and jaw). We denote the animation at frame t as $\mathbf{F}_t = [\mathbf{U}_t; \mathbf{L}_t]$, where $\mathbf{U}_t$ and $\mathbf{L}_t$ capture upper-area and lip-area motion, respectively.

3.2 Phone-Driven Lip Motion and Prosodic Expression Decomposition

The lip movements in facial animation are primarily driven by the sequential phone content of the utterance, forming a *phone-driven phonetic component*. In contrast, emotion-related expression variations that are not determined by phone sequences constitute the *prosodic component*.

Phone-driven component. We extract the phone-driven neutral lip motion $\mathbf{L}_t^n$ using an audio-driven facial animation generator (Daněček et al., 2023). Given expressive speech audio, we first obtain its transcription via speech recognition (Baevski et al., 2020), then synthesize pseudo-neutral speech (Google Cloud, 2026) that preserves phone content while removing expressive prosodic cues. The generator produces pseudo-neutral lip motion from this audio, serving as the phonetic baseline. Details of the extraction pipeline are provided in Appendix A.1.

Prosodic component. The prosodic expression offset is computed as $\mathbf{E}_t = \mathbf{L}_t - \mathbf{L}_t^n$, capturing emotion-related deviations from the phone-driven baseline. Temporal alignment between $\mathbf{L}_t^n$ and $\mathbf{L}_t$ is performed using the Montreal Forced Aligner (MFA) (McAuliffe et al., 2017) to ensure that offsets reflect genuine expressive deviations rather than artifacts of speech rate discrepancies. We explore two alignment strategies: Audio-Aligned (AA) and Motion-Aligned (MA); see Appendix A.2.

4 Methodology

LEMO, our proposed model, integrates a diffusion-based temporal inpainter and a region-aware ControlNet module, as illustrated in Fig. 1. The diffusion inpainter edits facial animation based on

personality attributes, while region-aware ControlNet enforces expressive lip-speech synchronization, conditioned on fine-grained region-state descriptors. This architecture enables precise, flexible editing, generating facial animations with desired expressions and natural speech articulation.

4.1 Problem Definitions

We cast the editing task as a controllable motion inpainting problem. Given a facial animation sequence $\mathbf{F} \in \mathbb{R}^{T\times D}$ where $D = D_u + D_l$, we decompose it into upper-area motion $\mathbf{U} \in \mathbb{R}^{T\times D_u}$, phone-driven neutral lip motion $\mathbf{L}^n \in \mathbb{R}^{T\times D_l}$, and prosodic expression offset $\mathbf{E} \in \mathbb{R}^{T\times D_l}$, where $\mathbf{E} = \mathbf{L} - \mathbf{L}^n$ and $\mathbf{L}$ denotes the full lip-area motion. Empirically, we set $D = 51$.

The model receives control signals that specify both personality attributes c (for Inpainter) and string embeddings $\mathcal{S}$ (e.g., blink details for ControlNet in Sec. 4.3). Personality attributes include the identity index $p_i \in [0, N-1]$, emotion label $p_e \in [0, M-1]$, and intensity $p_j \in [0, J-1]$, where N, M, J represent counts for persons, emotion labels, and intensity levels. At test time, these signals can be set manually to grant animators or LLMs direct control over the output.

A binary mask $m \in \{0,1\}^{T\times(D_u+D_l+D_l)}$ indicates which frames and facial areas are to be modified. Our aim is to inpaint the upper-area animation $\mathbf{U}$ and prosodic offset $\mathbf{E}$, thereby editing expression while preserving lip movement fidelity for speech alignment.

4.2 Lip Motion-Driven Inpainting

We adopt a temporal UNet architecture (Dhariwal & Nichol, 2021) trained using diffusion models to perform motion inpainting, conditioned on personality and unmasked input motion.

Following Ho et al. (2020), noise is added only to masked regions. Specifically, we define the input $x_0 = [\mathbf{U}, \mathbf{L}^n, \mathbf{E}] \in \mathbb{R}^{T\times(D_u+2D_l)}$. The forward diffusion process is:

$$\tilde{x}_k = \sqrt{\bar{\alpha}_k}x_0 + \sqrt{1-\bar{\alpha}_k}\epsilon, \qquad x_k = \tilde{x}_k \odot m + x_0 \odot (1-m), \tag{1}$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, k is the diffusion timestep, $\alpha_k = 1 - \beta_k$, $\bar{\alpha}_k = \prod_{i=1}^{k} \alpha_i$, and $\{\beta_k\}_{k=1}^{K}$ is a predefined variance schedule. We define personality condition as c.

The model is trained to reconstruct x_0 from masked noise using x_k, m, and c, leveraging the phone-driven neutral lip motion $\mathbf{L}^n$ and available expressive motion to recover masked regions $x_0 \odot (1-m)$. This process ensures consistency, naturalness, and adherence to specified personality attributes.

We closely follow Tevet et al. (2022b), training a network x_θ to predict the motion signal x_0 from the noisy x_k, minimizing the target loss:

$$\mathcal{L} = \mathbb{E}_{x_0,k,m,c}\left[\|x_0 - x_\theta(x_k, k, m, c)\|_2^2\right]. \tag{2}$$

At inference, noise is initialized in masked areas: $x_K = \epsilon \odot m + x_0 \odot (1-m)$, where K is the number of sampling steps; the reverse diffusion process then yields the final inpainted result x_0.

Masking Pattern and Schedule. To enable inpainter to effectively leverage correlations between masked and unmasked regions, preserving the underlying structure and temporal dynamics of facial animation, we incorporate a progressive masking schedule during training. This encourages the model to utilize context from unmasked regions for reconstructing missing information and to learn robust spatiotemporal relations.

In our masking scheme, the phone-driven neutral lip motion $\mathbf{L}^n$ is always kept unmasked to preserve the alignment between speech and lip articulation. The upper-area motion $\mathbf{U}$ and prosodic expression offset $\mathbf{E}$ are randomly masked according to a varying masking ratio r_{mask}. The change in masking ratio as training progresses enhances the model's ability to handle different levels of context loss. We empirically adopt the cosine growth masking scheme (see Appendix D.1).

Personality Injection. We condition facial animation inpainting on personality attributes by injecting identity index p_i, emotion label index p_e, and the corresponding emotion intensity p_j. A straightforward approach following EMOTE encodes personality attributes as one-hot vectors projected to a trainable style vector c. However, we found this causes training instability, likely because identity indices lack inherent semantic meaning.

To address this, we propose a more expressive embedding approach. We leverage a pre-trained text encoder (e.g., CLIP (Radford et al., 2021)) to map each combination of emotion label and intensity into a continuous embedding space, thus capturing their semantic relationships. The combination strictly follows a template: "<emotion>: <intensity>" (e.g., "happy: 0.7"). For identity, inspired by positional encoding in Vaswani et al. (2017), we use a fixed, non-trainable embedding table based on the sinusoidal encoding scheme, retrieving the identity embedding via index lookup. Both embeddings are then projected into a common feature space via independent linear layers and subsequently summed to form the final conditioning vector c.

4.3 Expression Editing with Region-aware ControlNet

While personality conditioning enhances the expressiveness of facial animation editing, it lacks the granularity needed for detailed, time-localized expression edits. For example, achieving a perfectly timed blink or precisely synchronized smile requires fine temporal and regional control. To address this, we propose a region-aware ControlNet.

Unlike conventional image-based ControlNets (Zhang et al., 2023) that manipulate local pixel values to achieve fine inpainting, the demands of facial animation editing are fundamentally different. Editing facial expressions must respect the complex structure of facial regions and the temporal continuity essential for believable animation. In addition, speech-driven sequences require strict preservation of articulation integrity, ensuring that expression modifications do not disrupt natural lip-speech synchronization.

To achieve this, our region-aware ControlNet injects context-aware and region-aware edit signals at every layer. These contextual control signals encode both explicit semantic intentions, such as requesting a blink or smile within a particular temporal window, as well as the current state of the animation. Region-aware ControlNet benefits from two specialized components: contextual translation for control representation, and attention-based adapter blocks for control signal injection, detailed as follows.

Contextual Translation for String Representation. Blendshape sequences possess inherent semantic structure, enabling intuitive mapping from continuous animation data to interpretable expression states. To facilitate high-level control of facial animation, we quantize blendshape values both spatially and temporally as follows:

1. Temporal Quantization. Every n-th frame is sampled to obtain $T' = \lceil T/n \rceil$ keyframes at timestamps $t_i = i \cdot n$, where $i = 0, 1, \ldots, T' - 1$. Each keyframe i governs a non-overlapping segment $\mathrm{seg}(i) = [\, in,\ \min((i{+}1)n - 1,\ T - 1)\,]$, so every frame belongs to exactly one segment; a trailing segment shorter than n frames is absorbed into the last keyframe. The interval n controls temporal resolution. **2. Spatial Quantization.** For each blendshape key $\mathrm{bs}_j \in \{1, \ldots, D_b\}$, K-means clustering with $\mathcal{K}$ centroids $\{\mu_{j,1}, \ldots, \mu_{j,\mathcal{K}}\}$ is applied to its activation values $B_j^{(t_i)}$. At keyframe k, $B_j^{(t_i)}$ is quantized as:

$$q_j(i) = \arg \min_{r \in \{1,\ldots,\mathcal{K}\}} \| B_j^{(t_i)} - \mu_{j,r} \|_2, \tag{3}$$

and mapped to an action category $\sigma_j(i)$ using a lookup table: $\sigma_j(i) \in \mathcal{A}_j$, where $\mathcal{A}_j$ is the discrete action set. [2]

Since each blendshape belongs to exactly one facial region, $\mathcal{T}_i$ further decomposes into per-region substrings $\{\mathcal{T}_{i,g}\}_{g \in \mathcal{G}}$, each collecting the descriptors of the blendshapes in region g. We embed each region substring independently with CLIP, obtaining a set of control tokens:

$$z_{i,g} = \mathrm{CLIP}(\mathcal{T}_{i,g}) \in \mathbb{R}^{D_z}, \qquad \mathcal{S} = \{z_{i,g}\} \in \mathbb{R}^{T' \times |\mathcal{G}| \times D_z}. \tag{4}$$

Each token is thus indexed by both keyframe i and region g, *enabling control signals region-aware and time-localized*.

[2] Each blendshape has an independent action set tailored to its anatomical meaning: For intensity-type blendshapes (e.g., JawOpen), $\mathcal{A}_j = \{$"low", "medium", "high"$\}$; for directional types (e.g., BrowInnerUp), $\mathcal{A}_j = \{$"left", "right"$\}$. See Appendix A.3 for more details.

The editing mask m is defined over blendshape channels (Sec. 4.1) and we lift it to the token space. The token-level mask aggregates the frame-level mask over the segment and channels of each token:

$$m^{\mathcal{S}}_{i,g} \;=\; \max_{t\in \mathrm{seg}(i)} \; \max_{d\in \mathrm{ch}(g)} \; m_{t,d} \;\in\; \{0,1\}, \tag{5}$$

i.e., a token is active if and only if its region is edited somewhere within its segment. The final control signal replaces inactive tokens with a learned null embedding ϕ and augments each token with a joint spatiotemporal positional encoding $\mathrm{PE}(i,g)$:

$$\tilde{\mathcal{S}}_{i,g} = m^{\mathcal{S}}_{i,g}\, z_{i,g} + (1 - m^{\mathcal{S}}_{i,g})\, \phi + \mathrm{PE}(i,g). \tag{6}$$

This quantization serves two key purposes. First, it reduces user burden by enabling high-level commands instead of frame-wise control. Second, it bridges with LLMs: each discrete state maps to interpretable tokens, allowing LLMs to process free-form editing requests and generate valid outputs. Crucially, $\mathcal{T}$ flows through two pathways: LLMs read, edit strings for intent reasoning, while ControlNet executes the corresponding embeddings $\tilde{\mathcal{S}}$ for motion synthesis.

Expression Adapter Blocks. We inject the control signal by inserting a cross-attention adapter into every UNet layer l: the hidden states $H_l \in \mathbb{R}^{T\times d_l}$ serve as queries (one per frame) and the control tokens $\tilde{\mathcal{S}}$ as keys and values, with a residual connection,

$$H_l^{\text{adapted}} = H_l + \mathrm{Attn}\big(H_l,\, \tilde{\mathcal{S}};\, M\big), \tag{7}$$

where Attn denotes standard scaled dot-product cross-attention. Two properties make the control frame-wise and region-specific: (i) every denoising frame issues its own query against tokens indexed by (keyframe, region), so the injected correction varies per frame while the token values carry region-specific content, with $\mathrm{PE}(i,g)$ aligning queries and tokens in time and space; (ii) the additive attention mask M formalizes the edit scope, with $M_{t,(i,g)} = -\infty$ for inactive tokens ($m^{\mathcal{S}}_{i,g} = 0$) and 0 otherwise, so unedited regions are excluded from the attention entirely and cannot perturb denoising outside the designated edit scope.

4.4 LLM-based Expression Editing

We leverage LLMs for intuitive, natural language-driven facial expression editing. Using carefully designed prompts for in-context learning (see Appendix B.4), LLMs interpret initial region-state descriptions and rewrite them based on user instructions. This enables flexible, fine-grained control, as LLMs effectively translate user intent into actionable edits. Consistent with ExpClip (Zhong et al., 2024), our findings show that LLMs accurately generate structured expression descriptors from arbitrary requests, facilitating seamless integration into the editing pipeline.

5 Experiments

5.1 Experimental Setup

We follow the experimental setup of recent audio-driven facial animation work (Daněček et al., 2023), using the MEAD dataset (Wang et al., 2020), a large-scale audio-visual corpus comprising 60 speakers with 8 emotions and 3 intensity levels. For high-fidelity 3D facial reconstruction, we employ EMOCA (Daněček et al., 2022), which predicts ARKit blendshapes and FLAME parameters from frontal facial video. The dataset is split in an 8:2 ratio for training and testing; additionally, identity and gender balances are maintained with roughly equal proportions in both splits. The maximum frame number of one facial animation sequence is set to 256.

We evaluate the proposed approach through two complementary experimental setups. The first is a high-level edit experiment that compares against existing baselines by manipulating emotion labels. This experiment consists of two sub-experiments: (1) the standard MEAD testing set evaluation, and (2) a MEAD-OpenVoice testing set evaluation that investigates the impact of audio content on generation methods. The second is a Vision-Language Model (VLM)-based fine-grained edit experiment that automatically assesses region- and time-specific control capabilities. We adopt Qwen3.7-Plus as the LLM and VLM. Implementation details are provided in Appendix B.

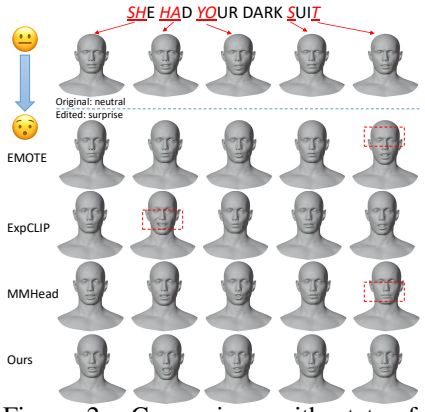


Figure 2: Comparison with state-of-the-art methods on expression editing, where LEMO achieves efficient modification with accurate lip-sync.

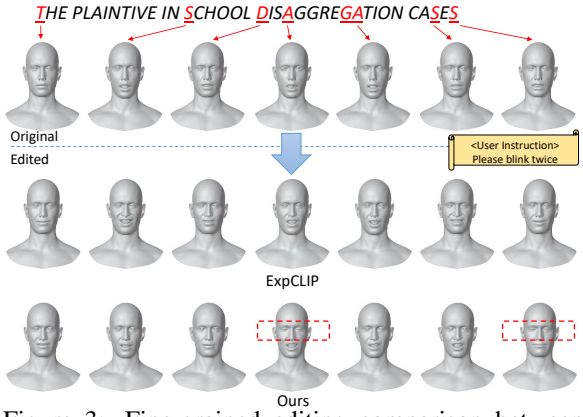


Figure 3: Fine-grained editing comparison between LEMO and ExpCLIP. Red boxes highlight successful edits.

Table 1: Quantitative comparison on the MEAD and our manipulated MEAD-OpenVoice testing set. **Bold** values denote the best performance and underlined values denote the second-best performance.

Method	MEAD					MEAD-OpenVoice				
	F-mAP ↑	S-mAP ↑	S-IoU ↑	LSE-D ↓	LSE-C ↑	F-mAP ↑	S-mAP ↑	S-IoU ↑	LSE-D ↓	LSE-C ↑
Ground truth	0.419	0.992	0.714	-	-	0.362	0.796	0.538	-	-
EMOTE (Daněček et al., 2023)	0.242	0.537	0.412	18.35	0.861	0.280	0.615	0.455	16.03	1.064
ExpCLIP (Zhong et al., 2024)	0.271	0.522	0.398	14.60	0.900	0.346	0.583	0.428	12.30	0.995
MMHead (Wu et al., 2024)	0.293	0.565	0.433	14.13	0.884	0.335	0.679	0.497	11.34	1.008
DEEPTalk (Kim et al., 2025)	0.352	0.579	0.448	10.90	1.107	0.433	0.700	0.524	8.96	**1.331**
3DiFACE (Thambiraja et al., 2025)	0.364	**0.683**	**0.521**	11.14	0.984	0.459	0.752	0.561	9.81	1.220
Ours	**0.398**	0.682	**0.521**	**9.83**	**1.112**	**0.465**	**0.766**	**0.573**	**8.80**	1.329

Baselines and Metrics. We compare LEMO with several state-of-the-art audio-driven and expression-editable facial animation methods, including: (1) EMOTE (Daněček et al., 2023): label-based editing of facial expressions, (2) ExpCLIP (Zhong et al., 2024): text-conditioned expression frames for whole-sequence generation, (3) MMHead (Wu et al., 2024): autoregressive facial animation from text, (4) DEEPTalk (Kim et al., 2025): VQ-VAE-based diffusion that distinguishes upper and lower facial regions, and (5) 3DiFACE (Thambiraja et al., 2025): diffusion inpainter that supports user-inserted keyframes for editing. As source code for some baselines (e.g., ExpCLIP, MMHead) is not publicly available, we re-implement these methods following the configurations reported in their respective papers and train all models with identical dataset splits.

For the high-level edit experiment, we use EMOTE's frame-wise and sequence-wise emotion classifiers to assess editing quality. Metrics include frame-wise (F-mAP) and sequence-wise (S-mAP) mean average precision, and sequence-wise Intersection over Union (S-IoU), which computes the proportion of correctly predicted target emotion frames using a threshold of 0.5. For lip synchronization, we report SyncNet metrics: Lip Sync Error Distance (LSE-D) and Confidence (LSE-C) following Aneja et al. (2023).

For the fine-grained edit experiment, we employ a VLM to automatically evaluate editing quality. The VLM receives the before-edit and after-edit animations alongside the original instruction, and rates execution success on a Likert scale from 1 (strongly disagree) to 5 (strongly agree). We report the average score across all testing samples.

Mesh Model for Visualization. For presentation and qualitative evaluation, we adopt a MetaHuman-based head mesh (Epic Games, 2025), professionally designed and fully compatible with ARKit blendshapes. Compared to generic FLAME meshes, MetaHuman heads exhibit finer facial details (e.g., wrinkles), enabling more expressive and realistic rendering of emotions and speaking styles (see Fig. 4). We offer a Blender add-on for convenient loading and visualization of animation sequences generated by our framework. FLAME-based motion parameters are converted to ARKit blendshapes using the EMAGE mapping (Liu et al., 2024).

Table 2: VLM-based fine-grained editing evaluation over 3 independent runs.

Method	Align. Score ↑	Win/Tie/Loss (%)	WR (%) ↑
ExpCLIP (Zhong et al., 2024)	3.78 ± 0.21	29.0 / 5.0 / 66.0	31.5 ± 2.1
Ours	$\mathbf{4.02 \pm 0.15}$	**66.0 / 5.0 / 29.0**	$\mathbf{68.5 \pm 1.8}$

5.2 Quantitative Evaluation

High-Level Edit Results. We conduct two experiments to evaluate high-level emotion editing. In the first experiment, we use the original MEAD testing set, replacing each sequence's emotion label with all eight emotional states to create eight variants per sequence. In the second experiment, we construct a MEAD-OpenVoice testing set using OpenVoice (Qin et al., 2023) to generate eight audio variants per utterance, each carrying a distinct emotional prosody while preserving phonetic content. This ensures prosody-emotion alignment while allowing utterance content to potentially conflict with the target emotion.

As shown in Tab. 1, on the original MEAD set, 3DiFACE achieves the highest S-mAP, but our method demonstrates superior F-mAP and lip synchronization quality. On MEAD-OpenVoice, where prosody is aligned with emotion, generation methods show improved performance but still achieve lower accuracy than our approach, confirming that audio content continues to influence their outputs. Overall, LEMO attains the best F-mAP, S-IoU and LSE-D on both testing sets, and performs on par with the strongest competitor on S-mAP and LSE-C.

This setup enables verification of the two failure modes described in the Introduction. **Failure Mode 1 (expression attenuation)** is assessed by comparing classification accuracy: generation methods consistently achieve lower F-mAP and S-mAP scores, indicating insufficient expression intensity. **Failure Mode 2 (emotion mismatch)** is demonstrated by the performance gap between testing sets. On MEAD-OpenVoice, generation methods improve when prosody is corrected but still underperform compared to our editing approach, confirming that prosody-phonetic entanglement persists in generation methods while our approach remains content-independent.

Fine-Grained Edit Results. Since most existing baselines condition solely on global emotion labels and cannot support region- or time-specific editing, we compare our approach only with ExpCLIP (Zhong et al., 2024), which supports text-conditioned expression control. We conduct a Vision-Language Model (VLM)-based fine-grained edit experiment that automatically assesses region- and time-specific control capabilities.

Specifically, we randomly select 100 sequences from the testing set. For each sequence, we use an LLM to generate fine-grained editing instructions in the form of string representations defined in the Contextual Translation subsection, saved as JSON files. Both our method and ExpCLIP then generate edited animations based on these instructions. The animations are rendered using standard methods (video examples are provided in supplementary). A VLM then evaluates the editing quality. We report the Alignment Score (mean VLM rating) and Win Rate (WR), in which ties count as half. The Wilcoxon signed-rank test is performed between Ours and ExpCLIP to assess statistical significance.

As shown in Tab. 2, our method achieves significantly higher alignment scores and win rates compared to ExpCLIP. This improvement can be attributed to the fundamental difference in editing paradigms. ExpCLIP synthesizes entire animations from scratch, making it difficult to precisely control specific facial regions or temporal intervals. In contrast, by separating phonetically driven lip motion from prosodic expression offsets, our method can selectively modify targeted facial components while preserving the rest of the animation, resulting in more accurate alignment with user-specified editing instructions.

Compositional Generalization. A central question is whether descriptor-level editing generalizes beyond the emotion-intensity combinations observed during training. We partition (region, action) descriptor combinations into a seen and an unseen set, where a combination is *unseen* if its (upper-region action, lip-region action) never appears in any training descriptor string. We retrain LEMO with all unseen combinations, and evaluate it under the same protocol. As reported in Tab. 3, editing unseen combinations degrades the alignment score only modestly. This indicates that the ControlNet learns local region-action responses whose composition transfers to co-occurrences absent from training, rather than memorizing the 24 emotion-intensity label combinations.

Table 3: Compositional generalization results on seen and unseen descriptor combinations.

Descriptor combinations	Align. Score ↑	WR (%) ↑
Seen	3.97 ± 0.16	81.3 ± 2.2
Unseen	3.86 ± 0.18	77.0 ± 2.5

Table 4: Ablation study of different training configurations in LEMO.

Variation	F-mAP ↑	S-mAP ↑	LSE-D ↓	LSE-C ↑
AA + Linear	0.351	0.576	8.49	0.958
AA + CLIP	0.353	0.658	9.53	0.980
MA + Linear	0.377	0.670	9.36	1.073
MA + CLIP (Final)	0.398	0.677	9.83	1.112

Human rating. To validate VLM evaluation, we also collect human ratings and compare them with VLM scores in Appendix D.2. Results show that our VLM evaluation serves as a reliable proxy for human assessment.

5.3 Qualitative Evaluation

In Fig. 2, we qualitatively compare LEMO against recent state-of-the-art methods. While all achieve satisfactory lip articulation synchronized with the spoken words, they exhibit limited capability in modifying facial expressions. For example, ExpCLIP produces an unnatural expression inconsistent with the intended "Surprised" emotion label; EMOTE's output appears stiff; and MMHead largely retains a neutral appearance. In contrast, our method accurately conveys the "Surprised" expression, as evidenced by the raised forehead wrinkles. This underlines the effectiveness of our phone-driven lip expression decomposition mechanism, which enables faithful and natural expression edits without sacrificing lip sync quality.

We further demonstrate fine-grained editing capabilities in Fig. 3. Given the user instruction "please blink twice", LEMO accurately executes two distinct blinks in the eye region at the specified temporal intervals while preserving lip synchronization. In contrast, ExpCLIP fails to interpret the semantic meaning of "blink" and only produces mouth movements, demonstrating its inability to perform region-specific temporal editing. This highlights the advantage of our LLM-guided approach: by converting continuous blendshape activations into discrete semantic descriptors, our method enables the LLM to reason about precise facial actions and generate structured editing instructions that our ControlNet can execute with spatial and temporal precision.

5.4 Ablation Study

We investigate the impact of different temporal alignment strategies and condition injection approaches mentioned through ablation studies. As shown in Tab. 4, "AA" denotes the alignment strategy with Audio Aligned, and "MA" denotes the alignment strategy with Motion Aligned. "Linear" denotes the condition injection approach proposed by EMOTE, and "CLIP" denotes our proposed approach using CLIP, which uses CLIP to encode template-fixed descriptions. The results indicate that the "MA + CLIP" combination yields the best performance, highlighting the necessity of capturing semantic relationships when injecting personality attributes. Furthermore, empirical evidence suggests that the motion alignment strategy is substantially more effective. Moreover, we conduct a user study with 40 participants comparing our method with EMOTE and ExpCLIP in Appendix C. Results show a clear user preference for our method, indicating superior performance.

6 Conclusion

In this paper, we addressed the challenge of expressive and controllable editing of speech-driven 3D facial animation. Rather than resolving prosody-phonetic entanglement at the generation stage, we propose LEMO, a motion-level editing paradigm for speech-driven facial animation. Our framework decomposes facial animation into phonetically driven lip motion and expressive offsets, then discretizes the latter into structured region-state descriptors. These descriptors enable LLMs to interpret free-form text instructions and guide a diffusion-based inpainter to perform precise region- and time-specific edits while preserving lip synchronization. Experiments on the MEAD dataset demonstrate that LEMO outperforms existing methods in expression fidelity and motion naturalness, while effectively circumventing the failure modes inherent in generation-based approaches.

AI USE STATEMENT

In this work, generative AI tools were used for two distinct purposes: text polishing and research evaluation. First, large language models were utilized solely to correct grammatical errors, refine sentence structures, and enhance the overall readability of the manuscript. Second, large language models (LLMs) and vision-language models (VLMs) were employed as evaluation aids within our user study. Specifically, an LLM was used to analyze qualitative user inputs, and a VLM was used to assess the quality of generated frame sequences.

We explicitly state that generative AI tools were not used for writing code, generating core research ideas, summarizing literature, or fabricating any primary scientific contributions of this paper. All AI-assisted evaluations and text revisions were rigorously reviewed and cross-checked word-by-word by the authors to guarantee that the original technical claims, scientific meaning, and experimental integrity remained completely unaltered. The authors take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the assistance of generative AI.

REFERENCES

Shivangi Aneja, Justus Thies, Angela Dai, and Matthias Nießner. Facetalk: Audio-driven motion diffusion for neural parametric head models. *arXiv preprint arXiv:2312.08459*, 2023.

Apple Developer. blendshapes — apple developer documentation. `https://developer.apple.com/documentation/arkit/arfaceanchor/blendshapes`, 2025. Accessed: 2026-03-13.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

Daniel Cudeiro, Timo Bolkart, Cassidy Laidlaw, Anurag Ranjan, and Michael J Black. Capture, learning, and synthesis of 3d speaking styles. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10101–10111, 2019.

Radek Daněček, Michael J Black, and Timo Bolkart. Emoca: Emotion driven monocular face capture and animation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20311–20322, 2022.

Radek Daněček, Kiran Chhatre, Shashank Tripathi, Yandong Wen, Michael Black, and Timo Bolkart. Emotional speech-driven animation with content-emotion disentanglement. In *SIGGRAPH Asia 2023 Conference Papers*, pp. 1–13, 2023.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Pif Edwards, Chris Landreth, Eugene Fiume, and Karan Singh. Jali: an animator-centric viseme model for expressive lip synchronization. *ACM Transactions on graphics (TOG)*, 35(4):1–11, 2016.

Epic Games. Metahuman creator. `https://www.metahuman.com/`, 2025. Accessed: 2026-03-16.

Yingruo Fan, Zhaojiang Lin, Jun Saito, Wenping Wang, and Taku Komura. Faceformer: Speech-driven 3d facial animation with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18770–18780, 2022.

Google Cloud. *Cloud Text-to-Speech*. Google, 2026. URL `google.com`. Accessed: 2026-02-01.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Liangxiao Hu, Hongwen Zhang, Yuxiang Zhang, Boyao Zhou, Boning Liu, Shengping Zhang, and Liqiang Nie. Gaussianavatar: Towards realistic human avatar modeling from a single video via animatable 3d gaussians. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 634–644, 2024.

Jisoo Kim, Jungbin Cho, Joonho Park, Soonmin Hwang, Da Eun Kim, Geon Kim, and Youngjae Yu. Deeptalk: Dynamic emotion embedding for probabilistic speech-driven 3d face animation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 4275–4283, 2025.

Zonglin Li, Xiaoqian Lv, Qinglin Liu, Quanling Meng, Xin Sun, and Shengping Zhang. Prosodytalker: 3d visual speech animation via prosody decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 5110–5118, 2025.

Haiyang Liu, Zihao Zhu, Giorgio Becherini, Yichen Peng, Mingyang Su, You Zhou, Xuefei Zhe, Naoya Iwamoto, Bo Zheng, and Michael J Black. Emage: Towards unified holistic co-speech gesture generation via expressive masked audio gesture modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1144–1154, 2024.

Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi. In *Interspeech 2017*, pp. 498–502, 2017. doi: 10.21437/Interspeech.2017-1386.

Ali Mollahosseini, Behzad Hasani, and Mohammad H Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE transactions on affective computing*, 10(1):18–31, 2017.

Ziqiao Peng, Haoyu Wu, Zhenbo Song, Hao Xu, Xiangyu Zhu, Jun He, Hongyan Liu, and Zhaoxin Fan. Emotalk: Speech-driven emotional disentanglement for 3d face animation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 20687–20697, 2023.

Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021.

Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20299–20309, 2024.

Zengyi Qin, Wenliang Zhao, Xumin Yu, and Xin Sun. Openvoice: Versatile instant voice cloning. *arXiv preprint arXiv:2312.01479*, 2023.

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*, pp. 8748–8763. PmLR, 2021.

Alexander Richard, Michael Zollhöfer, Yandong Wen, Fernando De la Torre, and Yaser Sheikh. Meshtalk: 3d face animation from speech using cross-modality disentanglement. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1173–1182, 2021.

Zhiyao Sun, Tian Lv, Sheng Ye, Matthieu Lin, Jenny Sheng, Yu-Hui Wen, Minjing Yu, and Yong-jin Liu. Diffposetalk: Speech-driven stylistic 3d facial animation and head pose generation via diffusion models. *ACM Transactions on Graphics (ToG)*, 43(4):1–9, 2024.

Sarah L Taylor, Moshe Mahler, Barry-John Theobald, and Iain Matthews. Dynamic units of visual speech. In *Proceedings of the 11th ACM SIGGRAPH/Eurographics conference on Computer Animation*, pp. 275–284, 2012.

Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. Motionclip: Exposing human motion generation to clip space. In *European Conference on Computer Vision*, pp. 358–374. Springer, 2022a.

Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022b.

Balamurugan Thambiraja, Malte Prinzler, Sadegh Aliakbarian, Darren Cosker, and Justus Thies. 3diface: Synthesizing and editing holistic 3d facial animation. In *2025 International Conference on 3D Vision (3DV)*, pp. 1462–1472. IEEE, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Kaisiyuan Wang, Qianyi Wu, Linsen Song, Zhuoqian Yang, Wayne Wu, Chen Qian, Ran He, Yu Qiao, and Chen Change Loy. Mead: A large-scale audio-visual dataset for emotional talking-face generation. In *European conference on computer vision*, pp. 700–717. Springer, 2020.

Sijing Wu, Yunhao Li, Yichao Yan, Huiyu Duan, Ziwei Liu, and Guangtao Zhai. Mmhead: Towards fine-grained multi-modal 3d facial animation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 7966–7975, 2024.

Jinbo Xing, Menghan Xia, Yuechen Zhang, Xiaodong Cun, Jue Wang, and Tien-Tsin Wong. Codetalker: Speech-driven 3d facial animation with discrete motion prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12780–12790, 2023.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023.

Yicheng Zhong, Huawei Wei, Peiji Yang, and Zhisheng Wang. Expclip: Bridging text and facial expressions via semantic alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7614–7622, 2024.

APPENDIX

A DATA PREPROCESSING DETAILS

This section provides additional details on the data preprocessing pipeline described in the Data Preprocessing section of the main paper, covering the phone-driven neutral lip motion extraction, temporal alignment strategies, K-means spatial quantization for the string representation, and the expression adapter block architecture.

A.1 PHONE-DRIVEN NEUTRAL LIP MOTION EXTRACTION

The phone-driven neutral lip motion extraction follows a three-step pipeline. Given expressive speech audio, the goal is to obtain a lip motion sequence that preserves the phonetic content (phone timing and articulation) while removing emotional prosodic cues.

- **Step 1: Speech Recognition.** The expressive speech audio is first transcribed using wav2vec 2.0 (Baevski et al., 2020) to obtain the spoken text.
- **Step 2: Pseudo-Neutral TTS Synthesis.** The transcription is fed into Google Cloud TTS (Google Cloud, 2026) to synthesize pseudo-neutral speech audio. This synthesized audio retains the phone timing of the original utterance while removing expressive prosodic cues, as TTS synthesis produces speech with near-neutral emotional tone by default.
- **Step 3: Audio-Driven Animation.** The pseudo-neutral speech audio is passed into EMOTE (Daněček et al., 2023), a pretrained audio-driven facial animation generator, to produce the pseudo-neutral lip motion $\mathbf{L}_t^n$. This motion serves as the phonetic baseline for subsequent decomposition.

A.2 TEMPORAL ALIGNMENT STRATEGIES

Temporal alignment between the pseudo-neutral lip motion $\mathbf{L}_t^n$ and the original expressive lip motion $\mathbf{L}_t$ is essential to ensure that the computed prosodic expression offset $\mathbf{E}_t = \mathbf{L}_t - \mathbf{L}_t^n$ captures genuine expressive deviations rather than artifacts induced by speech rate or phone duration discrepancies.

Table 5: Summary of spatial quantization configurations for the 45 enabled blendshapes.

Quantization Mode	K	Number of BS
K-means	2	19
K-means	3	16
Threshold	2	10
Total		45

We leverage the Montreal Forced Aligner (MFA) (McAuliffe et al., 2017), an open-source toolkit that produces precise start and end timings for each phone through phone-level alignment. We explore two alignment strategies:

- **Audio Aligned (AA).** The pseudo-neutral speech audio is generated to be temporally matched with the expressive audio. The animation generator then directly produces pseudo-neutral lip motion from this aligned audio. This strategy provides natural temporal alignment, but the phone durations of the TTS output may not perfectly match those of the original expressive audio.
- **Motion Aligned (MA).** The pseudo-neutral speech audio is first generated without temporal alignment constraints, and the corresponding pseudo-neutral lip motion is produced. MFA is then used to compute phone-level alignments between the pseudo-neutral and expressive audio. Based on the MFA timing information, time-warping interpolation is applied to align the pseudo-neutral lip motion with the expressive lip motion at the phone level. This strategy provides more precise phone-level alignment at the cost of potential interpolation artifacts.

Our ablation study demonstrates that the MA strategy substantially outperforms the AA strategy. We therefore adopt MA as the final alignment approach.

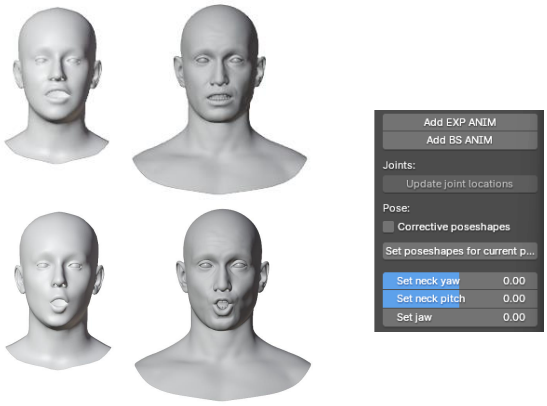
A.3 K-means Spatial Quantization Details

Out of the 51 ARKit blendshapes, 7 have activation values that are near zero across the entire training dataset and are therefore filtered out: "cheekPuff, cheekSquintLeft, cheekSquintRight, jawForward, noseSneerLeft, and noseSneerRight". The remaining 45 blendshapes are each independently quantized using one of two strategies, with the choice determined by the activation distribution characteristics of each blendshape.

- **K-means mode.** For blendshapes exhibiting a clear intensity gradient, K-means clustering is applied to the activation values across all training frames. Each blendshape is assigned K centroids, where $K \in \{2, 3\}$ is selected based on the distribution characteristics. With $K = 3$, the three centroids correspond to low, medium, and high activation levels. With $K = 2$, the two centroids correspond to low and high activation levels.
- **Threshold mode.** For blendshapes with a bimodal activation distribution, a fixed threshold of 0.5 is applied to partition the activation range into two clusters: inactive (< 0.5) and active (≥ 0.5).

Tab. 5 summarizes the quantization configuration for all 45 enabled blendshapes. The 10 threshold-mode blendshapes comprise 8 eye gaze blendshapes (eyeLookDownLeft/Right, eyeLookInLeft/Right, eyeLookOutLeft/Right, eyeLookUpLeft/Right) and 2 lip blendshapes (mouthPressLeft/Right). The 16 K-means blendshapes with $K = 3$ include brow blendshapes (browDownLeft/Right, browInnerUp, browOuterUpLeft/Right), eye blendshapes (eyeBlinkLeft/Right, eyeSquintLeft/Right), and mouth blendshapes (mouthFrownLeft/Right, mouthFunnel, mouthSmileLeft/Right, mouthStretchLeft/Right). The remaining 19 blendshapes use K-means with $K = 2$.

For each blendshape bs_j, the discrete cluster index $q_j(k)$ obtained from Eq.(7) of the main paper is mapped to a semantic action category $\sigma_j(k) \in \mathcal{A}_j$ according to a predefined lookup table derived



(a) Visual comparison (b) Blender Add-on Panel

Figure 4: Visual comparison of FLAME and our MetaHuman mesh: (a) Two aligned frames from the same animation sequence; (b) Our Blender Add-on interface for importing facial animations via blendshapes or FLAME parameters.

from the cluster centroid magnitudes. The action set $\mathcal{A}_j$ is tailored to each blendshape based on its anatomical function, following the examples described in the main paper.

The complete clustering rules, including centroid values, cluster statistics, and label mappings for all 45 blendshapes, are provided in the accompanying `cluster_rules.json` file.

A.4 MEAD-OpenVoice Testing Set Construction

To investigate the impact of audio content on generation methods, we construct a MEAD-OpenVoice testing set using OpenVoice (Qin et al., 2023), a zero-shot voice style cloning framework that enables emotion style transfer while preserving the original phonetic content.

OpenVoice achieves fine-grained tone color control by separating style features (e.g., emotion, rhythm, accent) from content features (phonetic information). Given a reference speech audio with a target emotion style and a source speech audio containing the desired phonetic content, OpenVoice synthesizes a new audio that combines the content of the source with the emotional prosody of the reference.

We apply OpenVoice to each utterance in the MEAD testing set as follows. For each original audio clip, we generate eight variants, each with a distinct emotional prosody corresponding to the eight emotion labels in MEAD (neutral, happy, sad, surprised, fear, disgusted, angry, contempt). The process preserves the phonetic content (phone timing and articulation) of the original utterance while replacing its emotional prosody with the target emotion style.

This construction ensures prosody-emotion alignment: the emotional prosody of each generated audio matches its assigned emotion label. However, the phonetic content may still reflect the original emotion, creating potential conflicts between utterance content and target emotion. This setup enables us to verify Failure Mode 2 (emotion mismatch): when phonetic content resembles a training sample with a different emotional label, generation methods tend to default to the training emotion rather than following the target emotion label. In contrast, our editing approach operates on pre-generated facial animations and is therefore content-independent, avoiding this entanglement.

B Implementation Details

B.1 Hyperparameters

The temporal downsampling interval is set to $n = 8$ frames. The initial mask ratio is $s_1 = 0.1$ and the maximum mask ratio is $s_2 = 0.9$. The diffusion inpainter is trained for 300 epochs and ControlNet for 100 epochs, with a batch size of 256 and the Adam optimizer. Training takes approximately 31 hours on a single RTX 4090 GPU. The text encoder is CLIP-ViT-B/32. During inference, we use

DDPM with 1000 diffusion steps. We tune these hyperparameters and present the evaluation results in the Additional Results section.

B.2 Emotion Classifier Details

We adopt the pretrained emotion classifiers provided by EMOTE (Daněček et al., 2023) for both frame-wise and sequence-wise emotion recognition. No additional training is performed. We describe each classifier below by rephrasing the relevant content from the EMOTE paper.

Frame-wise Classifier is EMOCA's publicly available emotion recognition network (Daněček et al., 2022), which predicts emotion features from a single image. The model consists of a ResNet50 backbone trained on AffectNet (Mollahosseini et al., 2017), producing a 2048-dimensional emotion feature for each frame. The network is trained to predict continuous valence and arousal values while simultaneously classifying eight emotion categories: neutral, happiness, sadness, surprise, fear, disgust, anger, and contempt.

Sequence-wise Classifier. While frame-level emotion features capture per-frame emotional signals, a single frame may conflate emotional expression with speech-related facial movements. To address this, EMOTE introduces a lightweight transformer-based sequence classifier that aggregates frame-level emotion features into a single video-level emotion classification.

The input video is passed frame by frame into the pretrained and frozen emotion recognition network, obtaining a 2048-dimensional emotion feature for each frame. These features are then passed into a single-layer transformer encoder to extract a single 256-dimensional video emotion feature. Instead of additive positional encodings, the model employs the ALiBi mechanism (Press et al., 2021), modified to also consider future frames. This feature is then fed into linear classification heads to classify the emotion and the intensity.

B.3 Expression Adapter Block Architecture

Adapter blocks are inserted at every UNet layer $l = 1, \ldots, L$. Each block performs cross-attention between the UNet hidden state and the masked string embeddings.

For layer l, let $H_l \in \mathbb{R}^{T \times d_l}$ be the UNet hidden representation before the adapter. Cross-attention projections use layer-specific trainable matrices:

$$Q_l = H_l W_Q^{(l)}, \quad K_l = \tilde{\mathcal{S}} W_K^{(l)}, \quad V_l = \tilde{\mathcal{S}} W_V^{(l)}, \tag{8}$$

where $W_Q^{(l)} \in \mathbb{R}^{d_l \times d_{att}}$, and $W_K^{(l)}, W_V^{(l)} \in \mathbb{R}^{D_z \times d_{att}}$. Here $\tilde{\mathcal{S}} \in \mathbb{R}^{T' \times D_z}$ is the masked-and-positionally-encoded string embedding from Eq.(10) of the main paper. Regions not being edited are filled with ϕ (CLIP null embedding), so their contribution to attention is zero-mean context.

Control injection is performed via scaled dot-product attention with a residual connection:

$$H_l^{\text{adapted}} = H_l + \text{Attention}(Q_l, K_l, V_l) \cdot W_O^{(l)}, \tag{9}$$

where $\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_{att}}}\right) V$ and $W_O^{(l)} \in \mathbb{R}^{d_{att} \times d_l}$ is the output projection. The total parameters per adapter layer amount to $d_l d_{att} + 2D_z d_{att} + d_{att} d_l$, which is significantly smaller than fine-tuning the full UNet.

B.4 Fine-Grained VLM Evaluation Details

As described in the Fine-Grained Edit Results paragraph of the main paper, we design a Vision-Language Model (VLM)-based evaluation protocol to automatically assess region- and time-specific control capabilities. The complete workflow involves four stages:

Stage 0: User Instruction Generation. For each of the 100 randomly selected test sequences, we first transcribe the speech audio into text using a speech recognition model. The transcribed text is then provided to an LLM. The LLM is instructed to imagine itself as a user who wants to modify the facial expression, and to generate a natural language editing instruction based on the spoken content. The complete prompt template for this stage is shown in Tab. 10.

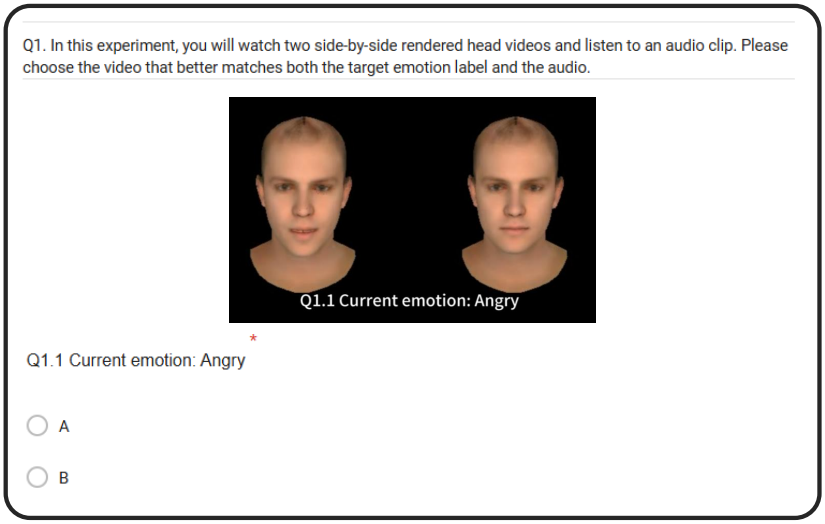


Figure 5: User interface for Q1: Participants view two side-by-side rendered head videos and an audio clip, then select the video that best matches both the target emotion label and the audio.

Stage 1: LLM-based Instruction Generation. The LLM receives the current facial state represented as string descriptors (as defined in the Contextual Translation subsection of the main paper) and interprets free-form user instructions to produce structured JSON outputs specifying which facial regions to modify and at what time intervals. The complete LLM prompt template is shown in Tab. 11.

Stage 2: Animation Editing. Both our method and ExpCLIP receive the JSON instructions and produce edited animations. Our method uses the LLM-generated instructions to guide the ControlNet for region-specific and time-specific editing, while ExpCLIP conditions on the textual descriptions for whole-sequence generation.

Stage 3: VLM-based Evaluation. A VLM evaluates the editing quality by comparing the before-edit and after-edit animations against the original JSON instructions. The VLM rates execution success on a Likert scale from 1 to 5. The complete VLM prompt template is shown in Tab. 12. We report the Alignment Score and Win Rate in the main paper.

C User Study

C.1 User Study Designs

We design two user studies to evaluate editing quality from different perspectives. The user interfaces are shown in Fig. 5 and Fig. 6, respectively.

- **Experiment 1: Binary Choice.** In this experiment, participants are presented with pairs of edited videos and asked to select the one that better conveys the target emotion. This binary forced-choice protocol measures relative preference. Each question pair consists of two videos produced by different methods from the same input, and participants select the method that better matches the target emotion label. The experiment contains 7 question pairs.
- **Experiment 2: Lip-sync Scoring.** In this experiment, participants rate the lip-sync quality of each edited video on a 1-to-5 scale, where 1 indicates the worst lip-sync quality and 5 indicates the best. This provides an absolute quality measure complementary to the relative preference of Experiment 1.

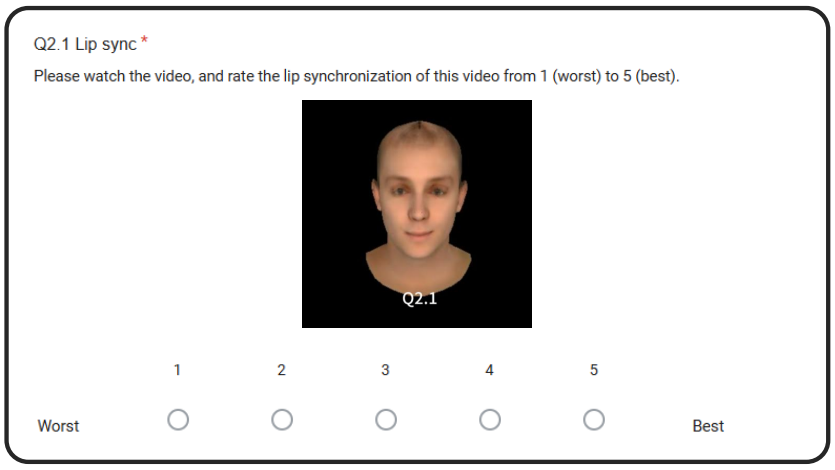


Figure 6: User interface for Q2: Participants watch a single rendered head video and rate its lip synchronization from 1 (worst) to 5 (best).

Table 6: Experiment 1 results: Binary choice preference rates. Participants selected the video that better conveys the target emotion.

Question	Emotion	Comparison	Winner
Q1.1	Angry	Ours vs. EMOTE	Ours (97.4%)
Q1.2	Happy	EMOTE vs. Ours	Ours (97.4%)
Q1.3	Surprised	ExpCLIP vs. EMOTE	ExpCLIP (79.5%)
Q1.4	Angry	EMOTE vs. ExpCLIP	ExpCLIP (94.9%)
Q1.5	Happy	ExpCLIP vs. Ours	Ours (71.8%)
Q1.6	Sad	Ours vs. ExpCLIP	ExpCLIP (94.9%)
Q1.7	Neutral	Ours vs. ExpCLIP	Ours (66.7%)

Video examples are selected from the MEAD testing set, covering multiple emotions and identities. Three methods are compared: our method, EMOTE, and ExpCLIP. All videos are anonymized and presented in random order to eliminate ordering bias. Examples are provided in the `visual_examples` directory of the supplementary material, with a detailed naming convention described in the accompanying `README.txt`.

C.2 Evaluation and Discussion

We recruit 40 graduate students as participants. All participants are given standardized instructions on the evaluation protocol before the experiment begins. The comparison methods include EMOTE (Daněček et al., 2023) and ExpCLIP (Zhong et al., 2024).

Experiment 1: Binary Choice. The results are shown in Tab. 6. Our method achieves an overall win rate of 80% in pairwise comparisons. Against EMOTE, Ours wins both comparisons with 97.4% preference in each case. Against ExpCLIP, Ours wins 2 out of 3 comparisons (71.8% for Happy, 66.7% for Neutral), while ExpCLIP wins the Sad comparison with 94.9% preference. Additionally, ExpCLIP consistently outperforms EMOTE in both comparisons (79.5% for Surprised, 94.9% for Angry). These results demonstrate that our editing approach is significantly preferred over EMOTE, while showing competitive performance against the generation-based method ExpCLIP.

Experiment 2: Lip-sync Scoring. The method-wise results are shown in Tab. 7. EMOTE achieves the highest mean lip-sync score, followed by our method, while ExpCLIP receives the lowest score.

Table 7: Experiment 2 results: Method-wise lip-sync quality scores (1-5 scale).

Method	Mean Score	Std
EMOTE (Daněček et al., 2023)	2.51	0.49
ExpCLIP (Zhong et al., 2024)	3.53	0.33
Ours	3.35	0.72

Additionally, we report statistical analysis: Wilcoxon signed-rank tests reveal that our method significantly outperforms EMOTE ($p < 0.0001$), while the difference between Ours and ExpCLIP is not statistically significant ($p = 0.1182$). ExpCLIP significantly outperforms EMOTE ($p < 0.0001$). These results indicate that:

1. Our method and ExpCLIP perform equally well on lip synchronization, with no significant difference ($p > 0.05$), meaning we lack sufficient evidence to reject the null hypothesis that both methods yield comparable lip sync quality. Although ExpCLIP has a slightly higher mean, this difference may be due to random variation, and we cannot conclude that ExpCLIP is truly better than Ours for lip synchronization.
2. Ours maintains better lip synchronization compared to EMOTE, supporting our core claim that editing pre-generated animations naturally preserves phonetic lip sync.

D Additional Results

D.1 Masking Pattern and Schedule

During training, the phone-driven neutral lip motion $\mathbf{L}^n$ is always kept unmasked to preserve speech-lip alignment, while the upper-area motion $\mathbf{U}$ and prosodic expression offset $\mathbf{E}$ are randomly masked. We experiment with two scheduling strategies:

1. Linear Growth: The masking ratio increases linearly with the number of training epochs (denoted as *lg*).
2. Cosine Growth: The masking ratio follows the cosine annealing schedule, defined as:

$$r_{mask}(e) = s_1 + \frac{1}{2}(s_2 - s_1)(1 - cos(\pi \cdot e/E)), \tag{10}$$

where e is the current epoch and E is the total number of epochs (denoted as *cg*).

We investigate two factors: the scheduling strategy (linear growth *lg* vs. cosine growth *cg*) and the masking ratio range (s_1, s_2). Cosine growth gradually accelerates then decelerates the masking ratio increase, whereas linear growth increases it at a constant rate. We evaluate four masking ratio ranges and report F-mAP scores in Tab. 8.

Table 8: Ablation study of masking scheduling strategies and masking ratio ranges (s_1, s_2). We report F-mAP scores on the MEAD dataset.

Strategy	$(0.1, 1.0)$	$(0.2, 1.0)$	$(0.1, 0.9)$	$(0.2, 0.9)$
lg	0.361	0.376	0.373	0.384
cg	0.355	0.387	**0.398**	0.381

The results show that cosine growth consistently outperforms linear growth for the $(0.2, 1.0)$ and $(0.1, 0.9)$ settings, suggesting that the non-linear schedule better encourages the model to exploit contextual information from unmasked regions. The best performance is achieved with cosine growth at $(s_1, s_2) = (0.1, 0.9)$, yielding an F-mAP of 0.398. We therefore adopt cosine growth with $(s_1, s_2) = (0.1, 0.9)$ as the default configuration.

Table 9: Agreement between VLM ratings and human ratings. Higher Spearman's ρ and Within-1 Accuracy, and lower MAE indicate better alignment with human perception.

Spearman's ρ	MAE	Within-1 Acc (%)
0.614	0.81	76.2

D.2 VLM-Human Agreement Study

To validate the reliability of our VLM-based evaluation protocol, we collect human ratings from 5 independent evaluators on the same set of 20 edited samples (our method results only). Each evaluator rates the videos using the same 5-point Likert scale. We then compute the agreement between VLM ratings and human ratings (aggregated as the mean across evaluators) using the following metrics:

- **Spearman's rank correlation (ρ):** measures whether the VLM and humans agree on the relative ranking of videos.
- **Mean Absolute Error (MAE):** measures the average absolute difference between VLM and human ratings.
- **Within-1 Accuracy:** the proportion of videos where the VLM rating differs from the human rating by at most 1 point.

As shown in Tab. 9, the VLM achieves a Spearman's correlation of 0.614 with human ratings, indicating moderate-to-strong agreement on the relative ranking of edited videos. The MAE suggests that on average, the VLM's ratings deviate from human ratings by less than 1 point. Furthermore, the Within-1 Accuracy indicates that in over three-quarters of cases, the VLM's rating falls within 1 point of the human consensus. These results confirm that our VLM-based evaluation protocol provides a reliable proxy for human judgment in assessing fine-grained expression editing quality.

E Limitations and Future Work

Limitations. Our method depends on source animations produced by a specific audio-driven generator (EMOTE). The editing quality is inherently constrained by the diversity and expressiveness of the source model. When the source animation lacks sufficient variation in a particular facial region, the editing headroom for that region is limited.

Future Work. We plan to extend the editing paradigm from 3D facial animation parameters to direct video editing, enabling application to real-world video content without requiring 3D morphable model fitting.

Table 10: LLM Prompt for user instruction generation.

Prompt #1: LLM Prompt for User Instruction Generation

```
You are a user who wants to edit a facial animation video.  You are
given the spoken text in the video:  "SPOKEN_TEXT"

Your task is to imagine how you would like to modify the facial
expression to better match the emotional tone of the spoken
content.

Your instruction should be:
- Natural and conversational, as if you are telling a video editor
what you want
- Specific enough to be actionable, but not overly technical
- Motivated by the spoken content (e.g., if the text expresses
surprise, you might want to raise eyebrows or widen eyes)

Please provide your editing instruction:
```

Table 11: LLM Prompt for Expression Editing.

Prompt #2: LLM Prompt for Expression Editing

```
You are a facial expression editing assistant. Your task is to
interpret the user's editing instruction and generate a structured
JSON output that specifies which facial regions to modify and when.

The facial animation system uses 45 blendshapes. Each blendshape
has been quantized into semantic action categories. Below is the
complete list:

Brow blendshapes (5):
- browDownLeft/Right: {relaxed, slightly_furrowed, furrowed}
- browInnerUp: {relaxed, slightly_raised, raised}
- browOuterUpLeft/Right: {relaxed, slightly_raised, raised}

Eye blendshapes (14):
- eyeBlinkLeft/Right: {relaxed, slightly_blinked, blinked}
- eyeSquintLeft/Right: {relaxed, slightly_tightened, tightened}
- eyeWideLeft/Right: {relaxed, wide}
- eyeLookDownLeft/Right, eyeLookInLeft/Right, eyeLookOutLeft/Right,
eyeLookUpLeft/Right: {off, on}

Mouth blendshapes (26):
- mouthSmileLeft/Right: {relaxed, slightly_smiling, smiling}
- mouthFrownLeft/Right: {relaxed, slightly_frowning, frowning}
- mouthFunnel: {relaxed, slightly_funneled, funneled}
- mouthStretchLeft/Right: {relaxed, slightly_stretched, stretched}
- jawOpen: {relaxed, open}
- jawLeft: {center, left}
- jawRight: {center, right}
- mouthLeft: {center, left}
- mouthRight: {center, right}
- mouthClose: {relaxed, closed}
- mouthDimpleLeft/Right: {relaxed, dimpled}
- mouthLowerDownLeft/Right: {relaxed, lowered}
- mouthUpperUpLeft/Right: {relaxed, raised}
- mouthPucker: {relaxed, puckered}
- mouthPressLeft/Right: {off, on}
- mouthRollLower: {relaxed, rolled}
- mouthRollUpper: {relaxed, rolled}
- mouthShrugLower: {relaxed, raised}
- mouthShrugUpper: {relaxed, raised}

Output format:
{"edits": [{"region": "<brow|eye|mouth>", "action":
"<action_category>", "time_range": [start_frame, end_frame]}]}

Rules:
1. Each edit must specify a facial region (brow, eye, or mouth).
2. The action must be one of the semantic categories listed above
for the specified blendshape.
3. The time_range specifies start and end frame indices.
4. If no time range is specified, assume [0, T-1] for the entire
sequence.
5. Multiple edits can be specified for different regions or time
ranges.
6. No explanation.

User instruction: "USER_INSTRUCTION"
```

Table 12: VLM Prompt for Fine-Grained Evaluation.

Prompt #3: VLM Prompt for Fine-Grained Evaluation

```
You are an expert in facial animation analysis.  You will be given:
1.  A "before-edit" facial animation video
2.  An "after-edit" facial animation video
3.  An editing instruction in JSON format

Your task is to evaluate how well the edited video executes the
intended editing instruction.

Rate the execution quality on a Likert scale from 1 to 5:
- 1 (strongly disagree):  The edit was not executed at all.
- 2 (disagree):  The edit was barely noticeable.
- 3 (neutral):  The edit was partially executed but does not fully
match the instruction.
- 4 (agree):  The edit was mostly executed with minor deviations.
- 5 (strongly agree):  The edit was perfectly executed, exactly
matching the instruction.

The editing instruction is:
JSON_INSTRUCTION

Please provide:
1.  A numerical score (1-5)
2.  A brief explanation
```