FacEDiT: Talking Head Video Editing via Facial Motion Infilling

Kim Sung-Bin[1] **Joohyun Chang**[2] **David Harwath**[3] **Tae-Hyun Oh**[2]

[1]POSTECH [2]KAIST [3]The University of Texas at Austin

Abstract

Editing a local segment of a talking head video without reshooting the scene remains challenging and underexplored. Local edits, such as insertion, deletion, or substitution, require synthesizing a replacement segment while leaving the rest of the video unchanged. This involves variable-duration local rewriting while preserving identity, unedited regions, and boundary continuity, which standard speech-driven generation and lip synchronization methods do not directly address. Moreover, direct supervision is impractical, as it would require paired videos of the same person and scene that differ only in a local spoken segment. We instead formulate talking head video editing as facial motion infilling, a self-supervised pretext task that recovers masked facial motion from speech and surrounding motion context using ordinary video–speech pairs. Our key insight is that local video editing can be simulated during training by masking a motion span and reconstructing it from speech and visible motion context. Building on this formulation, we introduce FacEDiT, a mask-controlled talking head model with a local temporal attention bias and temporal smoothness regularization to improve lip–speech alignment and transition continuity. We also introduce FacEDiTBench, the first benchmark for talking head video editing, covering diverse edit types and lengths with dedicated evaluation metrics. Extensive experiments show that FacEDiT produces accurate, speech-aligned edits with strong identity preservation and seamless boundary transitions. Beyond editing, the same facial motion infilling model extends to portrait animation and lip synchronization by simply changing the mask pattern, establishing FacEDiT as a unified framework for talking head video editing and generation. Code and benchmark data are available at https://facedit.github.io/.

1 Introduction

Filmmakers and media producers often need to revise specific parts of recorded videos, for example, when a word is misspoken or the script changes after shooting. In the iconic scene from Titanic (1997), Rose says, *"I'll never let go, Jack,"* but a director might later prefer *"I'll never let go no matter what happens, Jack."* Traditionally, such changes require reshooting the scene, which is costly and time-consuming. Recent progress in *talking head video synthesis* offers a promising alternative by generating or retargeting facial motion from speech. However, most existing methods are designed for speech-driven portrait animation or lip synchronization, rather than transcript-level edits that modify only a local video segment while leaving the remainder unchanged.

Concretely, talking head video synthesis is commonly studied in two settings (Fig. 1): *(a) speech-driven portrait animation*, which animates a reference image from speech, and *(b) speech-driven lip synchronization*, which modifies mouth motion in an existing video to follow target speech. In contrast, *(c) talking head video editing* performs local edits, such as substitution, insertion, and deletion, within an existing video. Although these tasks are closely related, talking head video editing poses distinct challenges: it must (i) synthesize a local replacement segment whose duration may

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

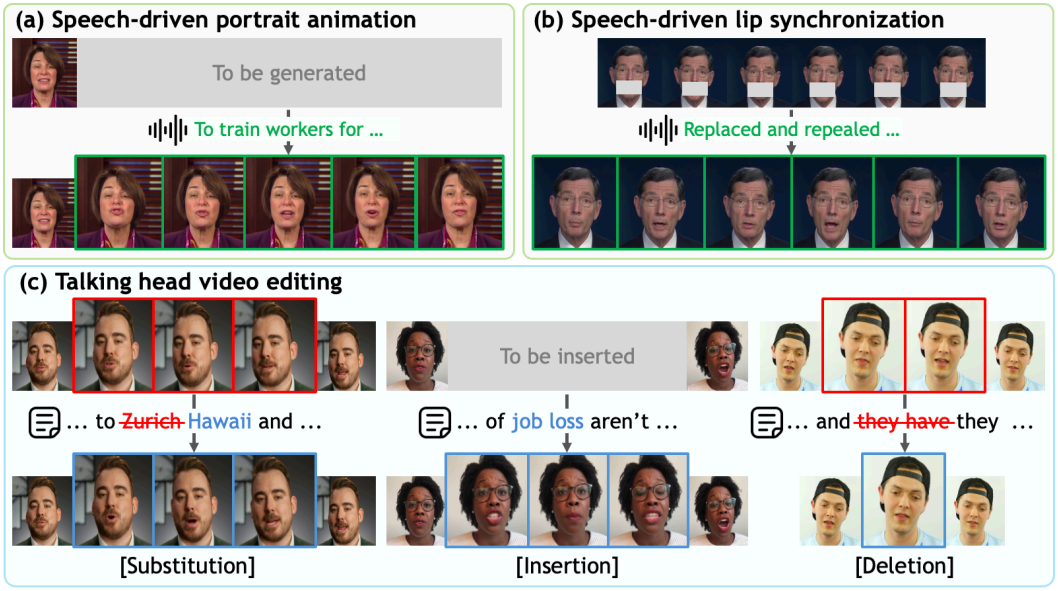


Figure 1: **Overview of FacEDiT.** FacEDiT performs mask-controlled facial motion infilling for local talking head video editing, regenerating only the edited span from speech and visible motion context. By changing the mask pattern, the same model supports portrait animation and lip synchronization.

differ from the original, (ii) preserve the original identity and all unedited regions, and (iii) maintain seamless transitions at the boundaries between edited and unedited frames.

Several talking head video editing approaches have been proposed, but they often require per-speaker optimization [1, 2] or degrade under long and complex edits [3]. A more fundamental obstacle is the lack of direct supervision, as paired videos of the same person and scene that differ only in a local spoken segment are impractical to obtain. As a result, speech-driven synthesis models are commonly repurposed as editing baselines, but their assumptions are mismatched. Portrait animation models [4, 5, 6, 7, 8, 9, 10, 11] generate full sequences rather than preserving unedited regions, so stitching locally generated segments back into the original video often introduces boundary discontinuities. Meanwhile, lip synchronization models [12, 13, 14, 15, 16] mainly retarget mouth motion under a driving pose trajectory, making them insufficient for duration-changing edits when driving content is unavailable. These limitations suggest that talking head video editing should be treated as a primary problem rather than a minor variant of speech-driven generation or lip retargeting.

In this work, we address talking head video editing through the lens of *facial motion infilling*. Given ordinary video–speech pairs, we construct a self-supervised pretext task by masking a temporal span of facial motion and training the model to recover the missing motion conditioned on speech and surrounding motion context. This converts editing from a supervised rewriting problem that requires paired edit data into a mask-conditioned reconstruction problem: the model learns to generate the edited region while using unmasked frames as context for identity preservation, motion continuity, and boundary consistency. Building on this formulation, we introduce **FacEDiT**, a mask-controlled talking head model that predicts compact facial motion latents from edited speech and visible motion context. To further improve lip–speech alignment and transition continuity, we incorporate a local temporal attention bias and temporal smoothness regularization.

Since no standard benchmark currently exists for talking head video editing, prior work lacks a common basis for evaluating whether an edit preserves identity and transitions smoothly across edit boundaries. To address this gap, we construct **FacEDiTBench**, a high-quality, human-verified dataset for this task, comprising 250 video and edited speech pairs. The dataset covers diverse edit types, edit lengths, and a wide range of identities. We also introduce dedicated metrics for identity preservation and boundary continuity to assess visual consistency and temporal smoothness in localized edits.

Extensive experiments show that FacEDiT enables accurate talking head video editing with precise lip synchronization, strong identity preservation, and seamless transitions, outperforming adapted

Table 1: **Comparison of talking head editing and synthesis settings.** Duration-changing local edit denotes modifying a local segment in an existing video while allowing the edited segment length to change, such as insertion or deletion. Unedited preservation indicates whether the method is designed to keep regions outside the edit unchanged. Boundary continuity refers to modeling smooth transitions between edited and unedited regions. Portrait animation and lip synchronization denote support for full-sequence speech-driven generation and mouth-motion retargeting, respectively.

Method / Setting	**Duration-changing local edit**	**Unedited preservation**	**Boundary continuity**	**Portrait animation**	**Lip sync**
Portrait animation [4, 19, 5, 7, 8, 20, 6, 10, 9]	No	No	No	Yes	Partial
Lip synchronization [12, 13, 14, 15, 16, 21, 22]	Partial	Partial	Partial	No	Yes
FacEDiT	Yes	Yes	Yes	Yes	Yes

baselines in quantitative and qualitative evaluations. Beyond editing, the same infilling strategy extends to related synthesis tasks by varying the mask pattern. Future-motion masking enables portrait animation, while lip-channel masking enables lip synchronization. This establishes FacEDiT as a unified framework for talking head editing and generation. Our contributions are summarized as:

- We cast talking head video editing as self-supervised facial motion infilling, learning variable-duration local edits from ordinary video–speech pairs without paired editing supervision.
- We propose FacEDiT, a mask-controlled motion infilling model that combines temporal attention bias and temporal smoothness regularization for speech-aligned edits and seamless transitions.
- We introduce FacEDiTBench, the first human-verified benchmark for talking head video editing, covering diverse edit types and lengths with dedicated identity and boundary-continuity metrics.
- We demonstrate that the same facial motion infilling model generalizes to portrait animation and lip synchronization through simple changes to the mask pattern, establishing FacEDiT as a unified model for talking head editing and generation.

2 Related work

Talking head video editing. Talking head video editing aims to modify only a local segment of an existing video according to transcript or speech edits, such as substitution, insertion, or deletion, while preserving the rest of the video. Early methods retrieve viseme–phoneme segments followed by neural rendering [1], perform iterative neural retargeting [2], or predict edited motion with person-specific neural rendering [3, 17]. These approaches often require target-speaker data or per-identity optimization, and their performance degrades for long or complex edits. Recent diffusion-based editing pipelines [18] further explore local editing, but remain task-specific and do not provide a general formulation that also covers related synthesis settings. Despite its practicality, talking head video editing remains underexplored due to limited standardized benchmarks and public implementations. To address this gap, we introduce FacEDiTBench, a high-quality, human-verified benchmark with dedicated metrics for talking head video editing.

Speech-driven synthesis as editing baselines. Speech-driven talking head synthesis is commonly studied as portrait animation or lip synchronization. Portrait animation generates a full talking head video from a reference image and speech, using explicit motion intermediates such as 3DMM coefficients [4] or landmarks [19], as well as recent diffusion and transformer-based generators [5, 7, 8, 20, 6, 10, 9]. Lip synchronization retargets mouth motion in an existing video to match target speech, using GAN-based or audio-visual supervision [12, 13], pose-disentangled representations [14], or diffusion-based inpainting and synchronization models [15, 16, 21, 22]. These models provide strong baselines for talking head video editing, but their assumptions are structurally mismatched with variable-duration local editing: portrait animation does not preserve fixed unedited frames, and lip synchronization assumes an available driving pose and focuses mainly on the mouth region.

Facial motion infilling for unified editing and generation. Recent talking head methods increasingly share compact motion representations, transformer backbones, and diffusion or flow-matching objectives, yet they remain tied to different task assumptions. As summarized in Table 1, existing settings do not jointly support duration-changing local edits, unedited-region preservation, boundary continuity, portrait animation, and lip synchronization. We instead formulate these settings as mask-controlled facial motion infilling, where temporal or channel-wise masks define the target task and

the model predicts masked facial motion from speech and visible motion context. This formulation directly addresses the supervision gap in editing while enabling a single model to support both editing and generation without task-specific architectures.

3 Method

We propose FacEDiT, a mask-controlled talking head video editing model based on self-supervised facial motion infilling. Given speech and visible motion context, FacEDiT learns to reconstruct missing facial motion within a masked temporal span, matching the editing setting where only a local segment is regenerated while neighboring frames remain fixed. The following subsections describe the infilling formulation, training objective, inference pipeline, and interpretation of our formulation.

3.1 Facial motion infilling formulation

We formulate talking head video editing as *facial motion infilling*. Given paired speech and video, we extract speech features $\mathbf{A} \in \mathbb{R}^{N \times D}$ and facial motion latents $\mathbf{F} = (\mathbf{f}_1, \ldots, \mathbf{f}_T) \in \mathbb{R}^{T \times C}$, where N and T denote the numbers of speech and video frames, and D and C denote the speech and motion feature dimensions. We sample a binary spatiotemporal mask $\mathbf{M} \in \{0,1\}^{T \times C}$, where $\mathbf{M}_{t,j} = 1$ indicates a motion entry to be predicted. Let $\mathbf{F}_{\mathbf{M}} = \mathbf{M} \odot \mathbf{F}$ denote the target motion entries and $\mathbf{F}_{\bar{\mathbf{M}}} = (1 - \mathbf{M}) \odot \mathbf{F}$ be the visible context. The goal is to learn the mask-conditioned distribution $p(\mathbf{F}_{\mathbf{M}} \mid \mathbf{F}_{\bar{\mathbf{M}}}, \mathbf{A})$ to infer missing facial motion from speech and visible motion context.

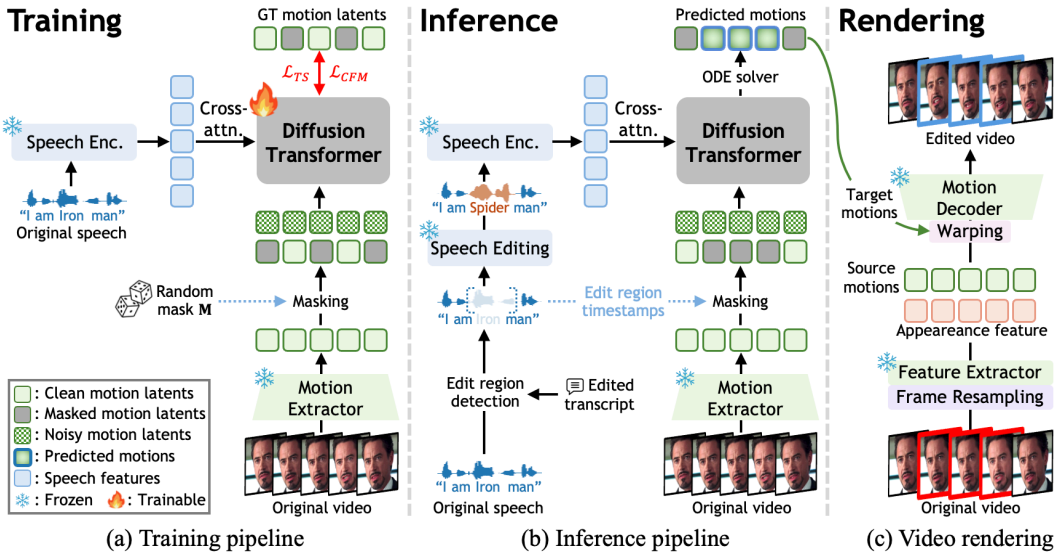
For talking head video editing, $\mathbf{M}$ covers the local span to be rewritten, so the model regenerates only the edited region while conditioning on surrounding unedited motion. The same formulation supports related synthesis tasks by changing the mask specification. For portrait animation, $\mathbf{M}$ masks all future motion after an observed initial prefix, allowing the model to complete the motion from speech. For lip synchronization, $\mathbf{M}$ masks only the lip-related channels while keeping pose and non-lip motion visible. Thus, a single model is controlled by the mask pattern at inference time.

3.2 Training FacEDiT

As illustrated in Fig. 2 (a), we train FacEDiT to solve the facial motion infilling objective in Sec. 3.1 using conditional flow matching (CFM) [23, 24]. Given ground-truth motion $\mathbf{F}_1$, Gaussian noise $\mathbf{F}_0$, and $t \sim \mathcal{U}[0,1]$, CFM defines the linear interpolation $\mathbf{F}_t = (1-t)\mathbf{F}_0 + t\mathbf{F}_1$ with target velocity $\mathbf{F}_1 - \mathbf{F}_0$. Given a sampled mask $\mathbf{M}$, the model is trained to reconstruct the target motion entries $\mathbf{M} \odot \mathbf{F}_1$ from noisy motion and the visible context $\mathbf{F}_{\text{cond}} = \mathbf{F}_{\bar{\mathbf{M}}} = (1 - \mathbf{M}) \odot \mathbf{F}_1$, conditioned on speech features $\mathbf{A}$. During training, we randomly sample the temporal location and length of the masked span, while the channel-wise mask follows one of two predefined patterns. Full-motion masking hides all facial motion dimensions within the selected temporal span, whereas lip-only masking hides only the lip-related channels while preserving the remaining expression, head rotation, and translation components [25]. This mixed masking strategy exposes the model to the mask patterns used across editing and related synthesis tasks.

Model architecture. We use the Diffusion Transformer (DiT) [26] as the backbone of our motion infilling model. The noisy motion and visible motion are concatenated along the channel dimension and projected into the DiT latent space, where convolutional positional embeddings are added. For speech conditioning, a straightforward approach is to concatenate the speech features with the noisy and visible motion features at the input level, referred to as early fusion. However, we empirically find that injecting speech features through multi-head cross-attention in each DiT block achieves better synchronization between lip movements and speech, as well as improved overall performance (see Table 5). We employ scaled rotary positional embeddings (RoPE) [27] to encode temporal ordering in both the self- and cross-attention layers.

Temporal multi-head attention bias. To encourage seamless transitions and precise lip-speech alignment, we incorporate a local temporal attention bias. Specifically, we add a bias $\mathbf{B}$ to the query–key attention logits in both self- and cross-attention, where $\mathbf{B}(i,j) = 0$ if $|i-j| < w$ and $-\infty$ otherwise. Given the projected query, key, and value features $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$, attention is computed as $\mathrm{softmax}(\mathbf{Q}\mathbf{K}^\top/\sqrt{d_k} + \mathbf{B})\mathbf{V}$, where d_k is the key dimension. This restricts each frame to attend to a local temporal window, encouraging the generated segment to connect smoothly with surrounding unedited motion and promoting local alignment between speech and motion features.



(a) Training pipeline (b) Inference pipeline (c) Video rendering

Figure 2: **Overview of FacEDiT**. (a) During training, FacEDiT learns facial motion infilling from paired speech and visible motion context using flow matching ($\mathcal{L}_{\text{CFM}}$) and temporal smoothness regularization ($\mathcal{L}_{\text{TS}}$). (b) During inference, the edit span or task setting defines the motion mask, and speech guides the masked motion prediction. (c) The predicted motion drives the source appearance features through the motion decoder to render the final video.

Learning objective. Our main objective is the mask-conditional CFM loss, which trains the model to predict the velocity field conditioned on speech, visible motion context, and the mask:

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{\mathbf{M}\sim q(\mathbf{M}),\, t,\, \mathbf{F}_0,\, \mathbf{F}_1,\, \mathbf{A}} \Big[\| \mathbf{M} \odot (v_\theta(\mathbf{F}_t, t, \mathbf{A}, \mathbf{F}_{\text{cond}}, \mathbf{M}) - (\mathbf{F}_1 - \mathbf{F}_0)) \|_2^2 \Big], \tag{1}$$

where $\mathbf{F}_t = (1-t)\mathbf{F}_0 + t\mathbf{F}_1$, $q(\mathbf{M})$ is the training mask distribution, and v_θ is our DiT-based model. To stabilize predicted motion dynamics and improve transition smoothness, we additionally apply a temporal smoothness loss $\mathcal{L}_{\text{TS}} = \frac{1}{T-1} \sum_{k=2}^{T} \| \hat{\mathbf{F}}_1^k - \hat{\mathbf{F}}_1^{k-1} \|_1$, where the clean motion estimate is obtained from the predicted velocity as $\hat{\mathbf{F}}_1 = \mathbf{F}_t + (1-t)\, v_\theta(\mathbf{F}_t, t, \mathbf{A}, \mathbf{F}_{\text{cond}}, \mathbf{M})$. The final objective is $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CFM}} + \lambda_{\text{TS}} \mathcal{L}_{\text{TS}}$, where λ_{TS} controls the smoothness weight.

3.3 Inference

Talking head video editing. Fig. 2 (b) shows our inference pipeline for talking head video editing. Unlike training, the model conditions on edited speech, and the mask is defined by the edit span. Given an original video-speech pair and an edited transcript, we use word-level forced alignment to localize the segment to be modified, and synthesize the edited speech with a speech editing model. We then extract facial motion latents from the original video, mask the localized segment, and adjust its duration to match the edited speech. Finally, the noisy motion, visible motion context, and edited speech are fed into the model to generate the final facial motion sequence.

Speech-driven portrait animation and lip synchronization. For speech-driven portrait animation, we use a future-motion mask that extends beyond the source motion to match the target speech duration. The model predicts the masked future motion with the available source motion and target speech features. For lip synchronization, we apply a lip-only channel mask while keeping head pose and expression dimensions unmasked; the remaining steps follow the editing procedure.

Video rendering and frame resampling. After predicting facial motion, we use it as the driving signal to animate the video. As shown in Fig. 2 (c), we extract appearance features and source motion from the original video, warp the source features using the predicted motion, and feed them into the frozen motion decoder to render the final edited frames while preserving identity. For duration-changing edits, the original frames must be resampled to match the edited speech length. To this end, we design a face-aware optical-flow-based resampling module that reduces artifacts from naive duplication, interpolation, or sparse sampling. Details are provided in Appendix C.3.

3.4 Interpretation of facial motion infilling

We provide a theoretical interpretation of FacEDiT as a mask-conditional generative model. The key observation is that different talking head synthesis tasks can be represented within a single model through different specifications of the mask $\mathbf{M}$.

Proposition 1 (Task unification via motion infilling)**.** *Talking head video editing, speech-driven portrait animation, and speech-driven lip synchronization can all be expressed as facial motion infilling problems by choosing different masks* $\mathbf{M}$*.*

For editing, $\mathbf{M}$ masks a local temporal span to be rewritten; for portrait animation, it masks future motion after an initial prefix; and for lip synchronization, it masks lip-related motion channels while keeping pose and non-lip motion visible. Thus, the same conditional distribution $p(\mathbf{F}_{\mathbf{M}} \mid \mathbf{F}_{\bar{\mathbf{M}}}, \mathbf{A})$ can represent all three task settings by changing only the mask specification.

Beyond the formulation-level view in Proposition 1, we show how random-mask training under conditional flow matching (CFM) enables a single model to amortize a family of mask-conditioned velocity fields and generalize across task masks at inference time.

Proposition 2 (Per-mask optimality under coverage)**.** *Let* $\mathcal{M}_{\text{task}}$ *be the set of masks required for tasks during inference. Under mild assumptions and sufficient model capacity, the optimal model obtained through random-mask training with the CFM objective in Eq.* (1) *recovers the Bayes-optimal conditional velocity field over the masked entries for each task mask* $\mathbf{M}^\star \in \mathcal{M}_{\text{task}}$*.*

This result suggests that, if the training mask distribution sufficiently covers the masks used at inference time, random-mask training allows FacEDiT to generalize across diverse task-specific masking patterns within a single conditional generation framework. More broadly, the unified formulation may extend beyond these three tasks, *e.g.*, to facial expression generation while keeping lip motion fixed. Nevertheless, the proposition should be interpreted as an idealized population-risk analysis rather than a guarantee for finite-capacity models. Detailed assumptions, discussion of the finite-capacity setting, and proofs are provided in Appendix E.

3.5 Implementation details

For facial motion representation and rendering, we adopt the frozen LivePortrait pipeline [28], which provides a compact per-frame motion representation and renderer, while FacEDiT predicts speech-conditioned masked motion sequences and their transitions to the visible context. Specifically, we use the expression deformation, head rotation, and translation components $[\delta, R, \tau] \in \mathbb{R}^{25\times 3}$, flattened into $\mathbf{f} \in \mathbb{R}^{75}$, corresponding to $C = 75$ in Sec. 3.1. This representation separates motion from appearance, allowing FacEDiT to modify facial dynamics while preserving fixed appearance features. Our model consists of 22 DiT layers with 16 attention heads and hidden/feed-forward dimensions of 1024/2048. VoiceCraft [29] is used for speech editing, while WavLM [30] encodes speech into D-dimensional features (D=768), which are projected into the DiT space. The LivePortrait motion and appearance extractors and decoder, as well as WavLM, are frozen; only FacEDiT's DiT and projection layer are trained.

For training, we use AdamW with a peak learning rate of 1e-4, linearly warmed up for 20K steps and decayed thereafter. We train on two A6000 GPUs with a total batch size of 8 for up to 1M steps with early stopping. The temporal smoothness weight λ_{TS} is set to 0.2. During inference, exponential moving average weights with a decay rate of 0.999 are used to stabilize predictions. Sampling uses the Euler ODE solver with timestep scaling based on the Sway strategy [31].

4 FacEDiTBench: Benchmark for talking head video editing

Dataset. To support thorough evaluation of talking head video editing, we construct FacEDiTBench, a first-of-its-kind dataset with 250 manually curated samples. Each sample includes an original video, original speech, original transcript, edited transcript, and edited speech. Videos are sourced from 100 samples from HDTF [32], 100 from Hallo3 [9], and 50 from CelebV-Dub [33], with the quality of both video and audio manually verified for each sample.

To construct the edited pairs, we prompt GPT-4o [34] to revise the original transcript to generate grammatically correct edits, and use VoiceCraft [29] with the original speech and edited transcript to

Table 2: **Statistics and example of FacEDiTBench.** (a) FacEDiTBench includes three types of edits with span lengths ranging from short to long. (b) An example of transcript editing shows a phrase insertion while preserving the surrounding content.

Edit span \ Edit type	Insertion	Substitution	Deletion	Total
1–3 words (short)	9	20	5	34
4–6 words (medium)	43	70	11	124
7–10 words (long)	29	59	4	92

(a) Statistics

Original transcript	Edited transcript
Father, it makes no sense to me.	Father, **I can't do that,** it makes no sense to me.

(b) Example of insertion edit type

synthesize the corresponding edited speech. Both the edited transcript and synthesized speech are manually verified for quality. For each sample, we annotate the type of modification, the start and end timestamps of the edited segment, and the editing span length. Specifically, the modification types include insertion, deletion, and substitution, while the editing span length varies from short (1–3 words) to medium (4–6 words) and long (7–10 words). FacEDiTBench does not include paired ground-truth edited videos, as obtaining near-identical recordings that differ only in a local spoken segment is impractical. Dataset statistics and an insertion example are shown in Table 2.

Metrics. Along with the dataset, we introduce metrics for two key aspects of editing quality: (1) boundary continuity, including photometric and motion continuity, and (2) identity preservation. Since ground-truth edited videos are unavailable and edited speech may correspond to multiple valid facial expressions, direct expression accuracy cannot be evaluated. We therefore focus on identity preservation and boundary continuity.

- **Photometric continuity ($\mathbf{P_{continuity}} \downarrow$)** measures pixel-level intensity changes at edit boundaries, indicating how naturally the color and lighting of the edited region blend with unedited areas.
- **Motion continuity ($\mathbf{M_{continuity}} \downarrow$)** measures variations in optical flow magnitude at edit boundaries using RAFT [35]. This metric captures the consistency of motion patterns across edited and unedited frames, reflecting temporal smoothness and stability in facial dynamics.
- **Identity preservation (IDSIM $\uparrow$)** computes cosine similarity between identity features of original and edited videos using a face recognition model [36].

5 Experiments

5.1 Experimental setup

Dataset. We train our model on a combination of video datasets totaling about 200 hours, including 130 hours from Hallo3 [9], 60 hours from CelebV-Dub [33], and 10 hours from HDTF [32]. Hallo3 and CelebV-Dub contain in-the-wild videos from diverse scenarios, such as movies and vlogs, while HDTF provides high-quality talking-head videos sourced from YouTube.

For talking head video editing evaluation, we use our proposed FacEDiTBench, which covers diverse editing scenarios and assumes that the speech has already been modified to match the edited transcript. We additionally evaluate speech-driven portrait animation and lip synchronization on the full HDTF test split, a standard benchmark for these tasks.

Competing methods. We compare FacEDiT with nine portrait animation models [4, 5, 20, 19, 6, 7, 8, 9, 10] and three lip synchronization models [16, 13, 15] using their official checkpoints, primarily on talking head video editing, and also on portrait animation and lip synchronization. Note that these models are not designed for the editing task; therefore, we use them only as baselines.[1]

Objective metrics. In addition to the metrics introduced in Sec. 4, we use standard measures of synthesis quality. Lip-sync error (LSE-D) and confidence (LSE-C) are computed with SyncNet [37], Fréchet Video Distance (FVD) [38] measures distributional video realism using real video segments as the reference distribution, and LPIPS [39] measures perceptual similarity. For editing, metrics are computed only on the edited segment; for portrait animation and lip synchronization, they are computed on the full sequence. All metrics are used across tasks except boundary continuity, which is specific to editing, and LPIPS, which is used only for portrait animation.

[1]Recent talking head video editing studies [17, 18], as well as earlier related works, do not release dedicated open-source models for this setting. We therefore repurpose existing portrait animation or lip synchronization models as practical baselines.

Table 3: **Comparison on talking head video editing.** We adapt speech-driven portrait animation and lip synchronization models as editing baselines (source task → editing). † uses boundary frames from the original video as keyframes for smooth interpolation while editing, and ∗ conditions on the original head pose and predicts only facial expression.

Method	LSE-D ↓	LSE-C ↑	IDSIM ↑	FVD ↓	$P_{continuity}$ ↓	$M_{continuity}$ ↓
Speech-driven portrait animation baselines → Editing						
V-Express [20]	7.719	6.381	0.845	201.392	40.668	11.470
AniPortrait [19]	10.601	0.904	0.905	141.574	8.445	8.281
EchoMimic [6]	9.313	4.289	0.885	158.631	12.924	17.892
EchoMimicV2 [10]	9.698	3.559	0.866	142.929	12.357	16.513
SadTalker [4]	7.644	6.202	0.851	112.951	8.243	7.442
Hallo [5]	8.244	5.664	0.909	99.454	7.242	6.429
Hallo2 [7]	8.205	5.724	0.918	95.256	7.598	6.001
Hallo3 [9]	8.754	5.559	0.880	106.525	7.651	7.311
KeyFace [8]	9.487	4.309	0.885	110.619	7.705	7.220
KeyFace† [8]	10.579	3.680	0.950	83.435	5.284	1.969
Speech-driven lip synchronization → Editing						
Wav2Lip [13]	7.202	**7.212**	0.950	83.118	4.369	1.180
LatentSync [16]	7.639	5.820	0.964	85.881	3.515	1.396
Diff2Lip [15]	8.139	4.453	0.936	71.818	3.473	1.686
SadTalker* [4]	7.852	5.847	0.852	98.893	7.713	6.803
AniPortrait* [19]	10.759	2.532	0.918	145.894	6.067	2.678
Ours	**7.135**	6.670	**0.966**	**61.930**	**2.420**	**0.800**

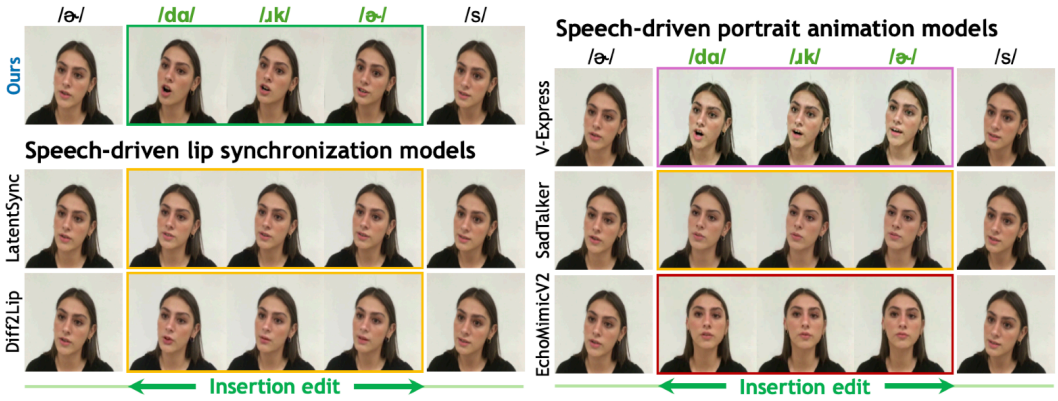


Figure 3: **Qualitative comparison.** The insertion edit adds the new word "darker" (/dɑɹkɚ/) to the original sentence. The green arrow indicates the edited region, while the others are original frames. Our model achieves accurate lip-sync, preserves identity, and transitions smoothly to the original frames. In contrast, V-Express fails to preserve identity; EchoMimicV2 shows a front-facing bias; SadTalker and lip synchronization models produce mumbled motion; and most methods introduce discontinuities at boundary transitions (*e.g.*, between /ɚ/ and /s/). Full results are in Appendix B.

5.2 Results and analyses

Comparison on talking head video editing. Since publicly available talking head video editing models are limited, we repurpose speech-driven portrait animation and lip synchronization models as editing baselines. Portrait animation baselines synthesize only the edited segment and stitch it back into the original video, while lip synchronization baselines resample original frames to the edited speech length and apply lip editing with extracted head pose. We also include pose-conditioned SadTalker and AniPortrait variants (∗), and a KeyFace extension (†) using original boundary frames as keyframes. As shown in Table 3, FacEDiT achieves the best boundary continuity and FVD while maintaining favorable lip-sync accuracy and identity preservation. Although KeyFace† and lip synchronization models improve boundary continuity over portrait animation baselines, they still lag behind FacEDiT, indicating that such adaptations only partially address local video editing.

Figure 3 shows a qualitative insertion example. Some baselines fail to preserve identity (V-Express), show a front-facing bias (EchoMimicV2), or produce mumbled motion (SadTalker and lip synchronization models), and most show boundary discontinuities at the transition between /ɚ/ and /s/. In contrast, our model preserves identity, produces smooth transitions, and generates lip motions consis-

Table 4: **Robustness to edit duration.** We evaluate 52 edits ranging from 10 to approximately 40 seconds, grouped by edit duration, where N denotes the number of edits in each group.

Edit duration	N	Avg. dur.	LSE-D ↓	LSE-C ↑	$P_{continuity}$ ↓	$M_{continuity}$ ↓	IDSIM ↑
10–20 s	18	15.89 s	7.419	6.061	1.259	0.360	0.9474
20–25 s	13	21.64 s	7.740	5.975	1.204	0.312	0.9469
25–30 s	9	26.93 s	7.582	5.583	1.772	0.388	0.9290
≥30 s	12	33.13 s	7.673	5.178	1.501	0.549	0.9200

Table 5: **Ablation results on FacEDiTBench.** We evaluate the impact of different design choices by comparing various configurations. "Early" and "CA" indicate the speech feature integration strategy (early fusion vs. cross-attention). (e) is the final model used in all experiments.

	Fusion	Speech Enc.	Biased Attn.	$\mathcal{L}_{TS}$	LSE-D ↓	LSE-C ↑	IDSIM ↑	FVD ↓	$P_{continuity}$ ↓	$M_{continuity}$ ↓
(a)	Early	WavLM	✓	✗	7.292	6.581	0.965	63.229	2.447	0.825
(b)	CA	WavLM	✓	✗	7.179	**6.673**	**0.966**	61.955	2.466	0.829
(c)	CA	Wav2Vec	✓	✗	7.223	6.639	0.965	62.171	2.448	0.821
(d)	CA	WavLM	✗	✗	7.301	6.655	0.955	65.559	2.816	1.097
(e)	CA	WavLM	✓	✓	**7.135**	6.670	**0.966**	**61.930**	**2.420**	**0.800**

tent with the edited speech. Overall, these results show that local video editing requires dedicated modeling beyond adapted speech-driven synthesis baselines.

Robustness to edit duration. To evaluate FacEDiT beyond the edit lengths covered by FacEDiT-Bench, we additionally evaluate 52 edits ranging from 10 to approximately 40 seconds, grouped by edit duration (Table 4). As the edit duration increases, we observe no sharp performance degradation, although lip-sync confidence and identity similarity gradually decrease, while motion continuity tends to degrade for longer edits. Based on these results, we conservatively regard approximately 30 seconds as the currently validated practical range, rather than a strict maximum.

Several edits beyond approximately 40 seconds could not be reliably evaluated because the upstream speech-editing model produced drifting speech. We exclude these cases because the failure occurs in the upstream speech-editing stage before facial-motion generation, rather than in FacEDiT itself. The relative robustness to long edits is consistent with our training setup, where long masked spans are predicted jointly rather than autoregressively.

Ablation study. We conduct ablations on FacEDiTBench to validate our design choices (Table 5). Comparing (b) with (a) and (c), cross-attention speech conditioning and WavLM [30] improve lip synchronization over early fusion and Wav2Vec [40], confirming the benefit of stronger speech conditioning. Removing the attention bias ((b) vs. (d)) noticeably degrades overall performance, especially lip synchronization and boundary continuity ($P_{continuity}$, $M_{continuity}$), highlighting its importance for speech alignment and smooth transitions. Adding the temporal smoothness loss $\mathcal{L}_{TS}$ in (e) further improves boundary continuity and reduces the RMS velocity and acceleration of predicted motion by 5.6% and 13.2%, respectively, further supporting its role in stabilizing motion dynamics (Appendix B). Based on these results, we use (e) as the final model configuration.

Generalization to portrait animation and lip synchronization. By changing mask patterns over facial motion latents, FacEDiT generalizes to portrait animation and lip synchronization. For portrait animation, we condition on the first frame and animate it with target speech; for lip synchronization, we follow the cross-generation setup [15], where a driving video from the same subject provides identity and head pose while target speech drives lip motion. As shown in Table 6, FacEDiT achieves favorable performance on both tasks: it improves lip-sync accuracy and realism in portrait animation, with lower LSE-D and FVD, and remains competitive with specialized lip synchronization methods. These results demonstrate that motion infilling not only excels at editing but also transfers effectively to related tasks within a unified framework.

Human evaluation. We further conduct a human study to assess perceptual quality across editing and generation tasks. For each comparison, 30 participants evaluate 20 samples by ranking the synthesized videos according to overall quality, including lip synchronization and motion naturalness. For talking head video editing, participants additionally evaluate boundary continuity between the edited and unedited regions. We conduct separate comparisons against portrait animation and lip synchronization models, and ranks are computed independently within each comparison pool. As shown in Table 7, FacEDiT is consistently preferred for talking head video editing in both overall

Table 6: **Results on cross-task generalization.** Our unified model generalizes to speech-driven portrait animation and lip synchronization, achieving favorable performance.

Method	LSE-D ↓	LSE-C ↑	IDSIM ↑	FVD ↓	LPIPS ↓
V-Express [20]	7.470	7.842	0.906	65.158	0.341
AniPortrait [19]	11.168	2.866	0.914	102.414	0.310
EchoMimic [6]	9.085	5.555	0.929	89.129	0.320
EchoMimicV2 [10]	9.966	4.051	0.890	131.603	0.322
SadTalker [4]	7.557	7.421	0.848	81.572	0.309
Hallo [5]	7.824	7.080	0.925	42.825	0.283
Hallo2 [7]	7.815	7.038	0.927	43.130	0.281
Hallo3 [9]	8.583	6.717	0.900	38.643	0.315
KeyFace [8]	8.905	5.795	0.926	49.043	**0.267**
Ours	**6.950**	**7.960**	**0.930**	**31.662**	0.289

(a) Portrait animation

Method	LSE-D ↓	LSE-C ↑	IDSIM ↑	FVD ↓
Wav2Lip [13]	7.433	7.635	0.931	42.843
Diff2Lip [15]	**6.910**	7.475	0.913	29.580
LatentSync [16]	7.177	7.533	0.933	**23.044**
Ours	7.132	**7.661**	**0.935**	29.381

(b) Lip synchronization

Table 7: **Human study comparison.** We report the average rank (± standard deviation) assigned by participants (lower is better) for talking head video editing and the corresponding generation task. Boundary continuity is evaluated for editing only. Ranks are computed independently within each comparison pool and are therefore not directly comparable across panels.

Task		Ours	SadTalker [4]	Hallo [5]	Hallo2 [7]	Hallo3 [9]	KeyFace [8]
Editing	Overall ↓	**1.57**±1.40	3.93±1.38	3.58±1.29	3.90±1.40	3.32±1.51	4.70±1.51
	Boundary ↓	**1.39**±1.25	4.19±1.29	3.35±1.07	3.90±1.34	3.23±1.49	4.93±1.39
Portrait animation	Overall ↓	**2.32**±1.60	4.83±1.50	3.04±1.35	2.91±1.41	3.68±1.59	4.23±1.47

(a) Comparison with speech-driven portrait animation models

Task		Ours	LatentSync [16]	Diff2Lip [15]	Wav2Lip [13]
Editing	Overall ↓	**1.22**±0.71	3.23±0.87	2.83±0.80	2.72±0.88
	Boundary ↓	**1.23**±0.67	3.19±0.92	2.74±0.77	2.83±0.95
Lip synchronization	Overall ↓	**1.31**±0.74	2.77±0.93	3.20±0.89	2.71±0.90

(b) Comparison with speech-driven lip synchronization models

quality and boundary continuity, with a clear margin over the competing methods. It also achieves favorable rankings on portrait animation and lip synchronization, showing that its favorable objective performance is also reflected in perceptual preference across tasks.

6 Conclusion and discussion

In this work, we presented FacEDiT, a mask-controlled framework that formulates talking head video editing as facial motion infilling. By masking the local edit span and reconstructing the missing facial motion from speech and surrounding motion context, FacEDiT enables variable-duration local edits while preserving identity, unedited regions, and boundary continuity. Beyond editing, the same infilling formulation extends to portrait animation and lip synchronization through future-motion and lip-channel masks, enabling a unified model for talking head editing and generation. We also introduced FacEDiTBench, the first dedicated benchmark for talking head video editing, with diverse edit types and dedicated metrics. Extensive experiments demonstrate accurate speech-aligned edits, strong visual consistency, and effective cross-task generalization, highlighting the potential of facial motion infilling for practical talking head video editing and synthesis.

While our method achieves strong performance in talking head video synthesis, it also has limitations. Since FacEDiT relies on the frozen LivePortrait motion extractor and renderer, it inherits their failure modes under extreme profile views, severe occlusions, and rapid head turns, which may lead to unreliable motion features or rendering artifacts. Occasional visual jumps and stale backgrounds may also remain due to the frozen pipeline, even when the predicted facial motion is plausible. Our model focuses on facial motion and therefore does not drive upper-body gestures from speech. In addition, emotional context is learned implicitly rather than modeled explicitly. Improving the motion extraction and rendering pipeline, extending the framework with modules for full-body motion, and incorporating explicit emotional cues could further improve realism.

Acknowledgments and disclosure of funding

Funding. This work was supported by the National Research Foundation of Korea (NRF) funded by the Korean government (MSIT) in 2026 (Grant No. RS-2024-00451947, Project Title: Interaction Framework for Connecting Users to Generative Experiences via Time-Space Expanded Multimedia: Centering on Social Communication Disorders (33%); Grant No. RS-2024-00453301, Project Title: MASC: Convergence of Multimodal AI and Semantic Communications for Hyper-Connected Hyper-Intelligent Public Safety Systems (27%)), the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. RS-2025-25441313, Professional AI Talent Development Program for Multimodal AI Agents (30%); No. RS-2026-25518317, Development of AI memory mechanism that reflects human cognitive principles (10%)), the InnoCORE program of the Ministry of Science and ICT (26-InnoCORE-01; N10250156), and the Artificial Intelligence Industrial Convergence Cluster Development Project funded by the Ministry of Science and ICT (MSIT, Korea) & Gwangju Metropolitan City.

Competing interests. The authors declare no competing interests.

References

[1] Ohad Fried, Ayush Tewari, Michael Zollhöfer, Adam Finkelstein, Eli Shechtman, Dan B Goldman, Kyle Genova, Zeyu Jin, Christian Theobalt, and Maneesh Agrawala. Text-based editing of talking-head video. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 2019. 2, 3

[2] Xinwei Yao, Ohad Fried, Kayvon Fatahalian, and Maneesh Agrawala. Iterative text-based editing of talking-heads using neural retargeting. *ACM Transactions on Graphics*, 2021. 2, 3

[3] Songlin Yang, Wei Wang, Jun Ling, Bo Peng, Xu Tan, and Jing Dong. Context-aware talking-head video editing. In *ACM International Conference on Multimedia (MM)*, 2023. 2, 3

[4] Wenxuan Zhang, Xiaodong Cun, Xuan Wang, Yong Zhang, Xi Shen, Yu Guo, Ying Shan, and Fei Wang. SadTalker: Learning realistic 3D motion coefficients for stylized audio-driven single image talking face animation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 3, 7, 8, 10, 17, 18, 22, 23, 24

[5] Mingwang Xu, Hui Li, Qingkun Su, Hanlin Shang, Liwei Zhang, Ce Liu, Jingdong Wang, Yao Yao, and Siyu Zhu. Hallo: Hierarchical audio-driven visual synthesis for portrait image animation. *arXiv preprint arXiv:2406.08801*, 2024. 2, 3, 7, 8, 10, 17, 22, 24

[6] Zhiyuan Chen, Jiajiong Cao, Zhiquan Chen, Yuming Li, and Chenguang Ma. EchoMimic: Lifelike audio-driven portrait animations through editable landmark conditions. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2025. 2, 3, 7, 8, 10, 17, 22

[7] Jiahao Cui, Hui Li, Yao Yao, Hao Zhu, Hanlin Shang, Kaihui Cheng, Hang Zhou, Siyu Zhu, and Jingdong Wang. Hallo2: Long-duration and high-resolution audio-driven portrait image animation. In *International Conference on Learning Representations (ICLR)*, 2025. 2, 3, 7, 8, 10, 17, 22, 24

[8] Antoni Bigata, Michał Stypułkowski, Rodrigo Mira, Stella Bounareli, Konstantinos Vougioukas, Zoe Landgraf, Nikita Drobyshev, Maciej Zieba, Stavros Petridis, and Maja Pantic. KeyFace: Expressive audio-driven facial animation for long sequences via keyframe interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2, 3, 7, 8, 10, 17, 23, 24

[9] Jiahao Cui, Hui Li, Yun Zhan, Hanlin Shang, Kaihui Cheng, Yuqi Ma, Shan Mu, Hang Zhou, Jingdong Wang, and Siyu Zhu. Hallo3: Highly dynamic and realistic portrait image animation with video diffusion transformer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2, 3, 6, 7, 8, 10, 14, 17, 18, 23, 24

[10] Rang Meng, Xingyu Zhang, Yuming Li, and Chenguang Ma. EchoMimicV2: Towards striking, simplified, and semi-body human animation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2, 3, 7, 8, 10, 17, 22

[11] Junuk Cha, Seongro Yoon, Valeriya Strizhkova, Francois Bremond, and Seungryul Baek. EmoTalking-Gaussian: Continuous emotion-conditioned talking head synthesis. *arXiv preprint arXiv:2502.00654*, 2025. 2

[12] K R Prajwal, Rudrabha Mukhopadhyay, Jerin Philip, Abhishek Jha, Vinay Namboodiri, and CV Jawahar. Towards automatic face-to-face translation. In *ACM International Conference on Multimedia (MM)*, 2019. 2, 3

[13] K R Prajwal, Rudrabha Mukhopadhyay, Vinay P Namboodiri, and CV Jawahar. A lip sync expert is all you need for speech to lip generation in the wild. In *ACM International Conference on Multimedia (MM)*, 2020. 2, 3, 7, 8, 10, 23, 24

[14] Hang Zhou, Yasheng Sun, Wayne Wu, Chen Change Loy, Xiaogang Wang, and Ziwei Liu. Pose-controllable talking face generation by implicitly modularized audio-visual representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 3

[15] Soumik Mukhopadhyay, Saksham Suri, Ravi Teja Gadde, and Abhinav Shrivastava. Diff2Lip: Audio-conditioned diffusion models for lip-synchronization. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2024. 2, 3, 7, 8, 9, 10, 23, 24

[16] Chunyu Li, Chao Zhang, Weikai Xu, Jingyu Lin, Jinghui Xie, Weiguo Feng, Bingyue Peng, Cunjian Chen, and Weiwei Xing. LatentSync: Taming audio-conditioned latent diffusion models for lip sync with SyncNet supervision. *arXiv preprint arXiv:2412.09262*, 2024. 2, 3, 7, 8, 10, 23, 24

[17] Benwang Chen, Xiaoyu Li, Xuan Wang, Qi Zhang, and Haoqian Wang. DFNeRF: Disentangled facial neural radiance fields for text-based editing of free-view talking head. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2025. 3, 7

[18] Bo Han, Heqing Zou, Haoyang Li, Guangcong Wang, and Eng Siong Chng. Text-based talking video editing with cascaded conditional diffusion. *arXiv preprint arXiv:2407.14841*, 2024. 3, 7

[19] Huawei Wei, Zejun Yang, and Zhisheng Wang. AniPortrait: Audio-driven synthesis of photorealistic portrait animation. *arXiv preprint arXiv:2403.17694*, 2024. 3, 7, 8, 10, 17, 22, 23

[20] Cong Wang, Kuan Tian, Jun Zhang, Yonghang Guan, Feng Luo, Fei Shen, Zhiwei Jiang, Qing Gu, Xiao Han, and Wei Yang. V-Express: Conditional dropout for progressive training of portrait video generation. *arXiv preprint arXiv:2406.02511*, 2024. 3, 7, 8, 10, 17, 22

[21] Ziqiao Peng, Jiwen Liu, Haoxian Zhang, Xiaoqiang Liu, Songlin Tang, Pengfei Wan, Di Zhang, Hongyan Liu, and Jun He. OmniSync: Towards universal lip synchronization via diffusion transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 3

[22] Junxian Ma, Shiwen Wang, Jian Yang, Junyi Hu, Jian Liang, Guosheng Lin, Jingbo Chen, Kai Li, and Yu Meng. SayAnything: Audio-driven lip synchronization with conditional video diffusion. *arXiv preprint arXiv:2502.11515*, 2025. 3

[23] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. 4, 26, 27

[24] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023. 4

[25] Mingi Kwon, Joonghyuk Shin, Jaeseok Jeong, Jaesik Park, and Youngjung Uh. JAM-Flow: Joint audio-motion synthesis with flow matching. In *European Conference on Computer Vision (ECCV)*, 2026. 4, 20

[26] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 4

[27] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024. 4

[28] Jianzhu Guo, Dingyun Zhang, Xiaoqiang Liu, Zhizhou Zhong, Yuan Zhang, Pengfei Wan, and Di Zhang. LivePortrait: Efficient portrait animation with stitching and retargeting control. *arXiv preprint arXiv:2407.03168*, 2024. 6, 17, 19, 20

[29] Puyuan Peng, Po-Yao Huang, Shang-Wen Li, Abdelrahman Mohamed, and David Harwath. VoiceCraft: Zero-shot speech editing and text-to-speech in the wild. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024. 6, 15

[30] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing (JSTSP)*, 2022. 6, 9

[31] Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, Jian Zhao, Kai Yu, and Xie Chen. F5-TTS: A fairytaler that fakes fluent and faithful speech with flow matching. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025. 6

[32] Zhimeng Zhang, Lincheng Li, Yu Ding, and Changjie Fan. Flow-guided one-shot talking face generation with a high-resolution audio-visual dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 6, 7, 14

[33] Kim Sung-Bin, Jeongsoo Choi, Puyuan Peng, Joon Son Chung, Tae-Hyun Oh, and David Harwath. VoiceCraft-Dub: Automated video dubbing with neural codec language models. In *IEEE International Conference on Computer Vision (ICCV)*, 2025. 6, 7, 14

[34] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 6, 15

[35] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, 2020. 7, 20

[36] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 7, 16

[37] Joon Son Chung and Andrew Zisserman. Out of time: Automated lip sync in the wild. In *Asian Conference on Computer Vision Workshops (ACCVW)*, 2016. 7, 23

[38] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 7, 24

[39] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 7, 24

[40] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 9

[41] Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. WhisperX: Time-accurate speech transcription of long-form audio. In *Conference of the International Speech Communication Association (INTERSPEECH)*, 2023. 15

[42] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 20

[43] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. MediaPipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*, 2019. 20

[44] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. 26, 28

Table of Contents

A Details on FacEDiTBench **14**

A.1 Dataset 14

A.2 Metrics 15

B Additional results and analysis **17**

C Additional implementation details **19**

C.1 Training 19

C.2 Inference 19

C.3 Frame resampling module 20

D Details on the experimental setup **22**

D.1 Comparison with speech-driven portrait animation models 22

D.2 Comparison with speech-driven lip synchronization models 23

D.3 Details on the objective metrics 23

D.4 Human evaluation 24

E Theoretical interpretation of facial motion infilling **25**

E.1 Task unification by mask specification 25

E.2 Mask-conditional flow matching 26

E.3 Proof of Proposition 2 26

F Applications and ethical considerations **28**

G Licenses and terms of existing assets **29**

H LLM usage **29**

A Details on FacEDiTBench

We introduce FacEDiTBench, a dedicated dataset and evaluation metrics for the talking head video editing task. Below, we describe the dataset construction pipeline and the evaluation metrics proposed for this task.

A.1 Dataset

As introduced in the main paper, FacEDiTBench contains diverse, high-quality video–speech pairs designed to evaluate talking head video editing. Each sample includes the original video and its corresponding speech, along with an edited transcript and edited speech for direct evaluation. The edited speech covers multiple edit types, including substitution, insertion, and deletion, with edit lengths ranging from one to ten words. Table S1 shows examples of transcripts for each edit type, which are later used to generate the edited speech. Human verification is incorporated at every stage to ensure dataset quality and accuracy.

Collecting video sources. We construct our dataset using videos from HDTF [32], CelebV-Dub [33], and Hallo3 [9]. CelebV-Dub and Hallo3 provide expressive, in-the-wild content across diverse sources, such as vlogs, dramas, and influencer videos, while HDTF offers high-quality interview

Table S1: **Examples of edit types in FacEDiTBench.** Each example shows the edit type and the number of edited words, along with the corresponding original and edited transcripts. The edited transcripts are used to generate edited speech for the dataset using VoiceCraft [29].

Edit type	Original transcript	Edited transcript
Substitution (5 words)	However, if **that naturally suits you**, then do	However, if **your heart tells you to**, then do
Insertion (6 words)	And I forget the story behind it, but it is really cool to see	And I forget the story behind it, **about how the festival first began**, but it is really cool to see
Deletion (6 words)	So we are not looking at **a murder, we are looking at** negligence	So we are not looking at negligence

videos from YouTube. Combining these datasets allows FacEDiTBench to cover a broad range of speaking styles, environments, and visual conditions.

Detecting English and labeling pseudo-transcripts. Since HDTF and Hallo3 do not include transcripts, we use WhisperX [41] to automatically detect the language and generate pseudo-transcripts. We retain only English-speaking videos and manually verify all generated transcripts for accuracy.

Generating edited transcript. For each verified transcript, we use GPT-4o [34] to generate diverse edited variants. We prompt GPT-4o with the original transcript to create multiple candidates covering different edit types (substitution, deletion, and insertion) and varying edit lengths (1–10 words). We then manually verify the edited transcripts and label each sample with its corresponding edit type and length. After this stage, each sample contains the original video with the corresponding speech, and the original and edited transcripts.

Generating edited speech. Using the original speech, original transcript, and edited transcript, we synthesize edited speech with VoiceCraft [29]. As VoiceCraft's outputs vary with hyperparameters (*e.g.*, duration or silence length), multiple samples are generated per edit, and we manually select the most natural and well-aligned version.

Human verification. Finally, all samples undergo a rigorous human verification process to ensure quality. Any samples with low-quality edited speech or noticeable artifacts are discarded, leaving only high-fidelity pairs. The resulting dataset contains 250 verified samples, covering diverse video sources, speaking styles, and editing scenarios.

A.2 Metrics

Along with the dataset, we introduce three evaluation metrics for the talking head video editing task. Specifically, we propose *photometric continuity* and *motion continuity* to assess boundary continuity between the edited and unedited regions, measuring whether the transition is smooth in terms of pixel appearance and motion dynamics. We additionally propose an *identity preservation* metric to evaluate whether the synthesized segment maintains the original identity.

Given an edited video, we evaluate how smoothly the edited region blends with the surrounding unedited frames. Using the example in Fig. S1, let the first to $(p-1)$-th frames and the $(q+1)$-th to the last frames be unedited, while frames p to q constitute the edited segment. For boundary continuity, we measure how naturally the video transitions across the boundaries at $(p-1)\rightarrow p$ and $q\rightarrow(q+1)$.

We illustrate the formulation using the right boundary as an example, although in practice boundary continuity is measured across all boundaries. We consider three consecutive frames q, $q+1$, and $q+2$, denoted by RGB images $I_q, I_{q+1}, I_{q+2} \in \mathbb{R}^{H\times W\times 3}$ with values in $[0, 255]$. The edit occurs between I_q and I_{q+1}, while I_{q+2} is an unedited frame serving as a natural reference. Using this setup, we define two continuity metrics, photometric continuity and motion continuity, which together assess whether the edited boundary introduces unnatural temporal or motion discontinuities.

Photometric continuity. This metric measures the second-order temporal derivative of pixel intensities across the boundary, capturing the photometric inconsistency introduced by the edited frame. Natural motion typically yields smooth temporal intensity trajectories, whereas abrupt or

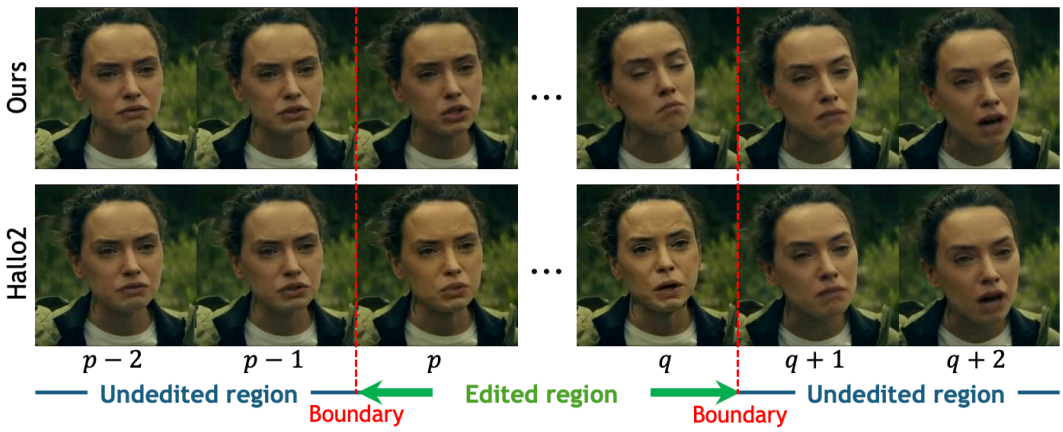


Figure S1: **Evaluation setup for boundary continuity.** Frames 1 to $(p-1)$ and $(q+1)$ to the last frame are treated as unedited, while frames p through q constitute the edited segment. Based on these segments, we compute the boundary continuity metric at the transition frames. Our model produces smooth transitions at these boundaries, whereas the comparison model (Hallo2) exhibits noticeable discontinuities, particularly between frames q and $q+1$.

artificial edits lead to large photometric acceleration. For each pixel and color channel, we compute

$$\mathrm{P}_{\text{continuity}} = \text{mean}\left(|I_{q+2} - 2I_{q+1} + I_q|\right),$$

which corresponds to the discrete second derivative across the sequence. High values indicate that the edited frame deviates from the natural photometric trajectory, whereas low values indicate temporally coherent appearance and smooth transition across the boundary.

Motion continuity. While photometric continuity captures intensity-level smoothness, motion continuity evaluates temporal consistency in the motion field using optical flow. Specifically, we examine whether the edited frame disrupts the expected symmetry between forward and backward motion around the boundary by measuring the change in optical-flow speed. We compute dense optical flow from the boundary frame to its neighboring frames as:

$$\begin{aligned} \text{Forward flow} &: v_{q+1\to q+2} && \text{from } I_{q+1} \text{ to } I_{q+2}, \\ \text{Backward flow} &: v_{q+1\to q} && \text{from } I_{q+1} \text{ to } I_q. \end{aligned}$$

We then compute their magnitudes:

$$s_f = \|v_{q+1\to q+2}\|_2, \qquad s_b = \|v_{q+1\to q}\|_2.$$

Temporal smoothness implies that the motion entering and exiting I_{q+1} should have similar magnitudes. Thus, we define the motion continuity as:

$$\mathrm{M}_{\text{continuity}} = \text{mean}\left(|s_f - s_b|\right).$$

Only flow vectors valid in both directions are considered. A low $\mathrm{M}_{\text{continuity}}$ indicates motion that transitions smoothly across the edited boundary, whereas a high value suggests unnatural acceleration or deceleration caused by the edit.

Photometric continuity evaluates temporal intensity smoothness, while motion continuity assesses motion-field consistency across the boundary. Together, they provide a comprehensive measure of whether the edited region blends seamlessly with the surrounding unedited content.

Identity preservation. Identity preservation measures how well the synthesized segment maintains the original identity. We compute cosine similarity between the averaged identity features of the original and generated videos using a face recognition model [36]. Given an original video segment of K frames $V_{1:K}$ and an edited video segment of T frames $\hat{V}_{1:T}$, we extract facial features

$$f_{1:K} = \text{FaceEnc}(V_{1:K}), \qquad \hat{f}_{1:T} = \text{FaceEnc}(\hat{V}_{1:T}),$$

Table S2: **Comparison on talking head video editing.** Note that the compared models are designed for speech-driven portrait animation and thus serve only as baselines, as they are not directly applicable to editing. We highlight the best results in **bold** and underline the second best.

Method	LSE-D (↓)	LSE-C (↑)	IDSIM (↑)	FVD (↓)	LPIPS (↓)
V-Express [20]	7.718	**7.264**	0.756	196.974	0.426
AniPortrait [19]	11.421	2.651	0.847	151.998	0.330
EchoMimic [6]	9.511	4.890	0.832	143.894	0.350
EchoMimicV2 [10]	10.136	3.819	0.817	124.219	0.347
SadTalker [4]	7.578	7.119	0.779	132.646	0.324
Hallo [5]	8.210	6.591	0.842	116.429	0.315
Hallo2 [7]	8.191	6.546	0.843	110.768	0.317
Hallo3 [9]	8.805	6.368	0.796	106.796	0.334
KeyFace [8]	9.258	5.289	0.796	129.082	0.307
Ours	**7.523**	7.218	**0.971**	**44.766**	**0.126**

Table S3: **Comparison of inference time.** We measure the inference time of each model when generating a talking head video from the same 15 seconds of speech.

Method	Inference (seconds)
Ours	73.787
V-Express [20]	639.694
AniPortrait [19]	280.817
EchoMimic [6]	808.065
EchoMimicV2 [10]	512.349

Method	Inference (seconds)
SadTalker [4]	177.455
Hallo [5]	1029.339
Hallo2 [7]	1056.977
Hallo3 [9]	2570.673
KeyFace [8]	372.119

where FaceEnc$(\cdot)$ denotes the face recognition model. We then compute the average feature vectors

$$f' = \text{mean}(f_{1:K}), \qquad \hat{f}' = \text{mean}(\hat{f}_{1:T}),$$

and define the identity similarity as

$$\text{IDSIM} = \text{cosine_similarity}(f', \hat{f}'),$$

which serves as the identity preservation metric.

B Additional results and analysis

We provide additional experimental results and analyses that are not included in the main paper due to space limitations.

Additional comparison with portrait animation models. In the main paper, we repurpose portrait animation baselines for talking head video editing by synthesizing only the locally edited region and stitching it back, which can introduce boundary artifacts. Here, we instead repurpose these baselines to generate the *entire* video from the edited speech. While full-sequence generation removes boundary discontinuities, it inevitably alters originally unedited regions, which is undesirable for localized editing. Accordingly, we evaluate both our method and the baselines over the full sequence.

Table S2 summarizes the quantitative results. Although lip-sync accuracy (LSE-D/C) improves slightly for most baselines compared to the main paper, identity preservation (IDSIM) degrades, indicating drift in the subject's appearance during full-sequence generation. We further compute LPIPS on the unedited portions by comparing the synthesized frames with the corresponding original frames. FacEDiT achieves the lowest LPIPS, showing better preservation of unedited content, whereas all baselines yield substantially higher LPIPS, confirming that portrait animation models fail to preserve the unedited regions and noticeably modify them. Overall, this experiment highlights the necessity of a dedicated talking head video editing model.

Inference time. Leveraging the efficient facial motion representation of LivePortrait [28], our model achieves substantially faster inference compared to prior work. Table S3 summarizes the inference time for generating a 15-second video on a single RTX A6000 GPU (for Hallo3, which requires more than 48GB of memory, we instead benchmark on an A100 GPU). Our model achieves up to *2.4*×

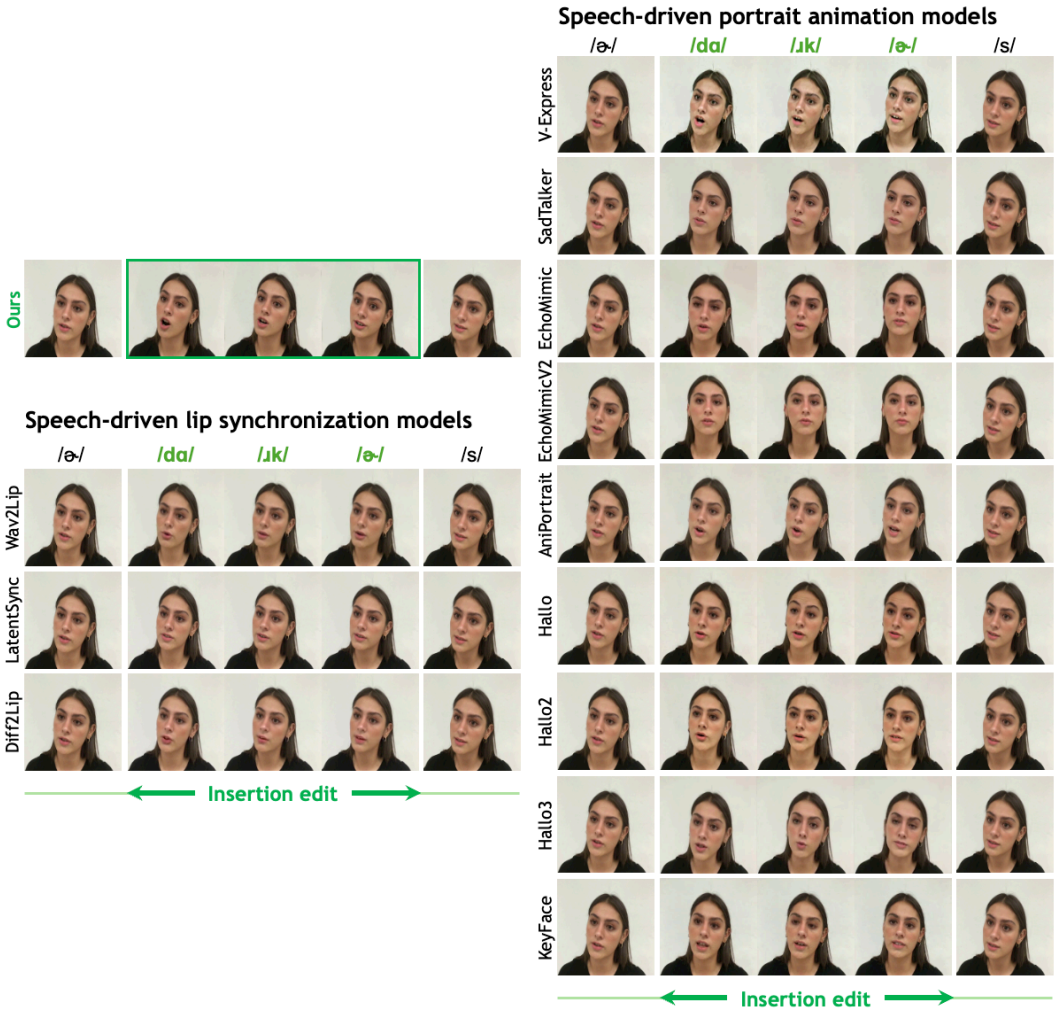


Figure S2: **Qualitative comparison.** Our method produces accurate lip synchronization with the edited speech while preserving identity and maintaining smooth transitions between edited and unedited frames compared to baseline models.

faster inference than the second-fastest portrait animation baseline, SadTalker [4], and up to *35*× *faster* than Hallo3 [9]. These results highlight not only the superior performance of our method in both objective and subjective evaluations, but also its remarkable efficiency, making it well-suited for both talking head video editing and speech-driven portrait animation.

Qualitative comparison. We showcase synthesized results from all models, including those partially shown in the main paper, on a talking head video editing sample (Fig. S2). Overall, our method produces accurate lip synchronization consistent with the edited transcript. Moreover, it preserves identity and maintains smooth transitions between edited and unedited frames, whereas the baseline models often result in blurred lip motion, boundary discontinuities, or identity drift. These results highlight the effectiveness of our approach for talking head video editing.

Analysis of temporal smoothness loss. We further analyze the effect of the temporal smoothness loss $\mathcal{L}_{\mathrm{TS}}$ on the predicted facial-motion trajectories. We compare the same ablation configurations with and without $\mathcal{L}_{\mathrm{TS}}$, and compute RMS velocity and acceleration from the first- and second-order temporal differences of the predicted motion latents over FacEDiTBench. As shown in Table S4, adding $\mathcal{L}_{\mathrm{TS}}$ reduces the RMS velocity from 0.001983 to 0.001872 (-5.6%) and the RMS acceleration from 0.001924 to 0.001671 (-13.2%). These reductions indicate smoother motion dynamics and are consistent with the improved boundary-continuity metrics in the main ablation study.

Table S4: **Motion smoothness analysis.** RMS velocity and acceleration of predicted facial-motion trajectories with and without the temporal smoothness loss.

Variant	Velocity ↓	Acceleration ↓
w/o $\mathcal{L}_{\mathrm{TS}}$	0.001983	0.001924
w/ $\mathcal{L}_{\mathrm{TS}}$	**0.001872**	**0.001671**
Relative reduction	5.6%	13.2%

C Additional implementation details

Facial motion representation. For facial motion representation, we adopt the motion latents from LivePortrait [28], a portrait animation framework that can animate a source image using a driving signal (*e.g.*, video or motion). For each image frame, the appearance feature h and the motion latent q are extracted through independent extractors. The motion latent is defined as $q = s \cdot (q_c R + \delta) + \tau$, where $q_c \in \mathbb{R}^{21\times 3}$ denotes canonical keypoints, $R \in \mathbb{R}^{3\times 3}$ represents head rotation, $\delta \in \mathbb{R}^{21\times 3}$ captures expression deformation, $\tau \in \mathbb{R}^{1\times 3}$ is translation, and s is a scale factor. Given source appearance features h_s, source keypoints q_s, and driving keypoints q_d, the warping module produces a warped feature that is decoded by the LivePortrait decoder to render the target animated frame.

Among the motion components, we use $[\delta, R, \tau] \in \mathbb{R}^{25\times 3}$ (concatenated across channels), which encode the essential facial dynamics for animation. We flatten these into a vector $f \in \mathbb{R}^{75}$ as our facial motion representation. This choice offers several practical advantages: (i) the representation is compact, structured, and easy to control; and (ii) the disentanglement between appearance and motion helps preserve identity, since the appearance feature remains fixed while only the relevant motion components are modified for talking head video synthesis.

C.1 Training

During training, we apply random masking to the facial motion sequence to perform motion infilling. Specifically, we randomly sample the temporal location and length of the masked span, covering 70%–100% of the sequence. The channel-wise mask follows one of two predefined patterns: a full-motion mask hides all facial motion channels within the selected span, whereas a lip-only mask hides only the lip-related channels while preserving the remaining expression, head rotation, and translation components. Thus, random masking refers to the temporal span, rather than to randomly selecting individual motion dimensions. For the attention bias, we use a window length of $w = 10$. The extracted speech features are at 50 Hz, while the video is at 25 FPS; therefore, we interpolate the facial motion features to 50 FPS so that the sampling rates of speech and facial motion are aligned.

C.2 Inference

As described in the main paper, our unified facial motion infilling formulation $p(\mathbf{F}_{\mathbf{M}} \mid \mathbf{F}_{\bar{\mathbf{M}}}, \mathbf{A})$, where $\mathbf{F}_{\mathbf{M}}$ denotes the masked motion, $\mathbf{F}_{\bar{\mathbf{M}}}$ denotes the visible motion, $\mathbf{A}$ denotes the speech feature, and $\mathbf{M}$ is the mask, can be instantiated for talking head video editing, speech-driven portrait animation, and speech-driven lip synchronization through task-specific mask definitions.

Talking head video editing. Let the edited segment correspond to the frame range $[t_1, t_2]$. The temporal mask is defined as:

$$M_t = \begin{cases} 1 & t_1 \le t \le t_2 \\ 0 & \text{otherwise} \end{cases}.$$

In this setup, the model infills the masked motion within the edited interval while conditioning on the surrounding frames and speech. Thus, editing is treated as a temporal motion infilling task.

Speech-driven portrait animation. Let the observed source motion be given up to frame T. The mask for speech-driven portrait animation is then:

$$M_t = \begin{cases} 0 & t \le T \\ 1 & t > T \end{cases}.$$

The model generates the subsequent motion sequence conditioned on the speech and the initial observed state. Consequently, portrait animation corresponds to sequence completion via motion infilling.

Speech-driven lip synchronization. Following [25], the LivePortrait motion latent can be decomposed into fine-grained components:

$$f_t = (f_t^{rest}, f_t^{lip}, f_t^{pose}, f_t^{trans}),$$

where f_t^{rest} represents non-lip facial expressions, and f_t^{lip}, f_t^{pose}, and f_t^{trans} correspond to lip motion, head pose, and translation, respectively. For speech-driven lip synchronization, we define a channel-wise mask:

$$M_{t,rest} = 0, \quad M_{t,lip} = 1, \quad M_{t,pose} = 0, \quad M_{t,trans} = 0.$$

Under this configuration, the model reconstructs lip motion conditioned on speech and the unmasked motion components while preserving the remaining parameters. Thus, lip synchronization is formulated as a channel-wise motion infilling task.

C.3 Frame resampling module

A key challenge is that the edited region may differ in length from the original segment, requiring frame resampling before final video rendering. Naive duplication or interpolation for upsampling can introduce artifacts, while downsampling often causes severe aliasing. To address this, we design a face-aware resampling module based on dense optical flow and soft interpolation, which adjusts the RGB frame sequence to the edited duration while preserving coherent motion and visual continuity. The resampled frames, together with the motion predicted by FacEDiT, are then passed to the LivePortrait decoder [28] to render the final edited video.

Given an input sequence of N frames, we generate M output frames by retaining the first and last frames and synthesizing the remaining $(M-2)$ intermediate frames. Let I_0 and I_L denote the endpoints, and let $t \in (0,1)$ be the normalized timestamp of an intermediate frame. We synthesize I_t by (1) estimating bidirectional flows between I_0 and I_L, (2) warping both endpoints toward time t, and (3) blending the warped images using a softmax-weighted forward-splatting formulation [42]. This endpoint-based strategy reduces aliasing during downsampling and avoids abrupt frame repetition during upsampling.

Endpoint flows. First, we estimate the dense optical flow between the endpoints using RAFT [35] as:

$$F_{0\to L} = \text{RAFT}(I_0, I_L), \qquad F_{L\to 0} = \text{RAFT}(I_L, I_0).$$

To approximate motion at time t, we scale the flows as

$$F_{0\to t} = t\, F_{0\to L}, \qquad F_{L\to t} = (1-t)\, F_{L\to 0},$$

and warp I_0 and I_L accordingly to obtain I_0^t and I_L^t.

Soft forward–backward reliability. To account for occlusions and inaccurate flow, we compute soft forward–backward (FB) consistency weights:

$$w_0 = \exp(-\beta\, \text{FB}(F_{0\to L}, F_{L\to 0})),$$
$$w_L = \exp(-\beta\, \text{FB}(F_{L\to 0}, F_{0\to L})),$$

where $\text{FB}(x) = \|F_{a\to b}(x) + F_{b\to a}\big(x + F_{a\to b}(x)\big)\|_2$, with x a pixel coordinate, measures the forward-backward flow inconsistency. A small floor value is applied to avoid zero weights and ensure stable blending.

Face-aware reliability (MediaPipe). Because talking head videos exhibit structured facial motion, we incorporate a face-aware spatial prior to emphasize reliable regions during blending. We use MediaPipe Face Mesh [43] to detect dense facial landmarks on I_0 and I_L, extract the facial contour, and rasterize it into binary face masks ROI_0 and ROI_L. These masks amplify reliability within the face region:

$$w_0' = w_0\big(0.8 + 0.2\,\text{ROI}_0\big), \qquad w_L' = w_L\big(0.8 + 0.2\,\text{ROI}_L\big),$$

encouraging smooth motion around the face.

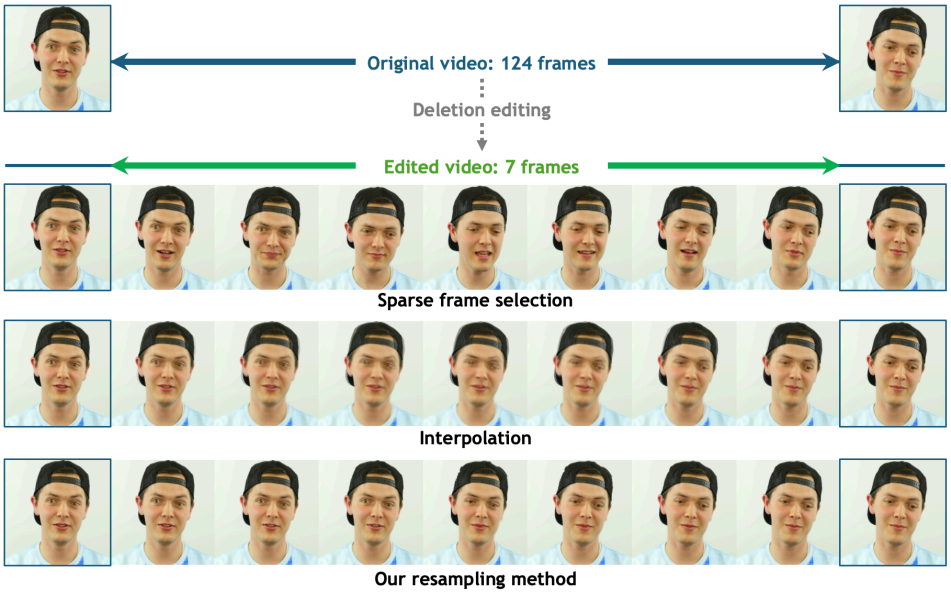


Figure S3: **Qualitative comparison of frame resampling methods.** When downsampling a video (*e.g.*, from 124 frames to 7), an effective resampling strategy is required. Sparse frame selection introduces aliasing, while interpolation between only the first and last frames produces noticeable blur. In contrast, our resampling method produces smoother transitions with significantly fewer artifacts.

Softmax forward splatting. The intermediate $(M-2)$ frames are synthesized via softmax-weighted forward splatting of the warped endpoints:

$$\alpha_0 = (w'_0)^\gamma (1-t), \qquad \alpha_L = (w'_L)^\gamma t,$$

$$I_t = \frac{\alpha_0 \cdot \text{splat}(I_0^t) + \alpha_L \cdot \text{splat}(I_L^t)}{\alpha_0 + \alpha_L + \varepsilon},$$

where γ is the softmax sharpness exponent for blending and $\text{splat}(I)(x) = \sum_y I(y)\,\omega(y \to x)$, with $\omega(y \to x)$ denoting the bilinear weight by which pixel y contributes to location x after being warped by the flow. Forward splatting distributes each source pixel to the target frame using flow-aligned bilinear weights, providing dense coverage even under large motion.

Hole-aware fallback. If the accumulated splat weight falls below a threshold, we fall back to the nearest endpoint warp.

$$I_t(x) = \begin{cases} I_0^t(x), & t < 0.5, \\ I_L^t(x), & t \geq 0.5. \end{cases}$$

Result. The resampled sequence

$$[I_0, I_{t_1}, I_{t_2}, \ldots, I_{t_{M-2}}, I_L]$$

contains all interior frames synthesized via face-aware softmax forward splatting. This produces temporally smooth, identity-consistent, and hole-free transitions, which are crucial for the later stages of facial animation rendering.

Figure S3 compares different downsampling strategies for reducing 124 frames to 7. A simple baseline is sparse frame selection, which samples frames at evenly spaced intervals; however, this causes strong aliasing and abrupt temporal changes. Another approach is linear interpolation between the endpoint frames, but pixel-space interpolation across such a long temporal gap introduces noticeable blur. In contrast, our resampling module produces smooth motion with far fewer artifacts. These observations are supported by Table S5, where we report quantitative results on editing samples that require downsampling. We find that our resampling method improves lip-sync accuracy as well as boundary continuity compared to both sparse sampling and interpolation-based approaches.

Table S5: **Quantitative comparison of resampling methods.** Using our frame resampling approach during downsampling improves lip-sync accuracy and boundary continuity compared to sparse frame selection and interpolation-based methods.

Method	LSE-D (↓)	LSE-C (↑)	IDSIM (↑)	$P_{continuity}$ (↓)	$M_{continuity}$ (↓)
Sparse selection	7.769	5.137	**0.968**	3.609	1.587
Interpolation	7.716	5.215	0.954	3.139	0.829
Our resampling	**7.336**	**5.588**	0.960	**3.118**	**0.825**

D Details on the experimental setup

D.1 Comparison with speech-driven portrait animation models

We compare against nine speech-driven portrait animation models on both the talking head video editing and portrait animation tasks. For the editing setting, we provide only the edited speech segment (excluding unedited portions) to each model, generate the corresponding video segment, and stitch it back into the original video. Note that these models are designed for portrait animation rather than editing, and thus serve only as baselines for the editing task.

V-Express [20]. V-Express adopts a progressive training strategy that balances strong and weak control signals. A latent diffusion backbone is extended with conditioning modules for reference images and audio cues, while V-Kps pose inputs are encoded by a lightweight guider and injected directly into the latent space. Conditional dropout prevents over-reliance on strong conditions, enabling effective audio-driven motion. This design produces identity-consistent, pose-aligned, and audio-synchronized portrait videos.

AniPortrait [19]. AniPortrait maps audio to 3D facial structures and projects them into 2D landmarks, which serve as a robust intermediate representation for a diffusion-based generator. By conditioning on these audio-driven landmarks and a reference portrait image, the model synthesizes temporally coherent frames with strong identity preservation. The explicit 3D geometry offers controllable motion and enables natural expressions as well as applications, such as facial motion editing.

EchoMimic [6]. EchoMimic jointly leverages audio cues and facial landmarks within a unified diffusion framework. Audio and landmark features are encoded separately and injected into the denoising process, allowing the model to operate with audio only, landmarks only, or both. This dual-conditioning setup stabilizes motion prediction under weak audio signals while maintaining geometric consistency informed by landmarks. The resulting animations preserve identity and offer fine-grained control of facial motion.

EchoMimicV2 [10]. EchoMimicV2 extends generation to the half-body setting through a streamlined conditioning strategy. An audio–pose dynamic harmonization module reduces dependence on detailed pose inputs while broadening audio control from local lip motion to full-body dynamics. Minimal structural cues, such as a single-frame hand-pose estimate, guide expressive upper-body movements. Phase-specific denoising objectives further improve stability. For fair comparison, we provide hand pose extracted from the first frame using the authors' implementation.

SadTalker [4]. SadTalker estimates 3DMM motion coefficients from audio, including head pose and expression parameters. The predicted motion is transformed into an unsupervised 3D keypoint space and rendered through a 3D-aware generator for strong identity preservation. By modeling the relationship between audio and various motion components, the method produces stable lip movements and expressive head dynamics with controllable animation.

Hallo [5]. Hallo uses a hierarchical audio–visual cross-attention mechanism that aligns audio features with specific facial regions. Region-specific masks for lips, expressions, and head pose are applied within the same latent resolution, enabling fine-grained motion control while maintaining reference-image appearance. This hierarchical structure enhances temporal coherence and motion diversity while preserving identity.

Hallo2 [7]. Hallo2 employs incremental motion synthesis with patch-drop augmentation to improve long-range consistency. Motion tokens are predicted autoregressively and aligned with audio cues through a transformer backbone. Patch-drop increases robustness to missing temporal infor-

mation, while hierarchical cross-attention fuses audio, motion, and spatial features into a unified representation.

Hallo3 [9]. Hallo3 uses a video diffusion transformer that models global spatiotemporal relationships across frames. Appearance cues from the reference image, audio-driven motion signals, and text-guided modulation are integrated to produce expressive, identity-preserving facial dynamics. Because Hallo3 requires a text prompt that indicates the appearance at inference time, we use a fixed neutral prompt ("A person talking") for all comparisons.

KeyFace [8]. KeyFace enhances temporal coherence via a keyframe-driven synthesis approach. In stage 1, a sparse set of key motion frames is predicted, followed by stage 2 interpolation of intermediate frames using a U-Net denoiser conditioned on audio features. This keyframe–interpolation design stabilizes identity, reduces temporal drift, and produces smooth, lip-synced motion over long sequences. Progressive conditional dropout further enhances robustness during training. In the main paper, we also report a KeyFace variant: boundary frames from the original video (corresponding to the edit boundaries) are used as stage-1 keyframes, enabling stage-2 interpolation to better preserve boundary continuity than the original setup.

D.2 Comparison with speech-driven lip synchronization models

We compare three speech-driven lip synchronization models on both talking head video editing and lip synchronization, and additionally include two variants adapted from portrait animation models as lip synchronization baselines. For editing, we resample the original video to match the edited speech length, use it as the driving video, and modify the lower-face region according to the edited speech. As these models are designed for lip synchronization, they serve only as editing baselines.

LatentSync [16]. LatentSync is an end-to-end lip-sync framework built on an audio-conditioned latent diffusion model that operates in the latent space of Stable Diffusion, avoiding explicit intermediate motion representations such as landmarks or 3D models. To improve temporal consistency across generated frames, it introduces TREPA (Temporal REPresentation Alignment), which aligns temporal representations of generated and ground-truth frame sequences extracted by a large-scale self-supervised video model, serving as an additional temporal regularizer without adding model parameters.

Diff2Lip [15]. Diff2Lip formulates lip synchronization as an audio-conditioned diffusion inpainting problem over the mouth region. It takes three inputs: (i) a masked input frame to provide pose/context, (ii) a reference frame to provide identity and mouth texture, and (iii) an audio frame/window to drive lip shape; the diffusion model then generates the lip-synced mouth region conditioned on these cues.

Wav2Lip [13]. Wav2Lip uses a CNN encoder–decoder generator with three components: an identity encoder (encoding a reference frame together with a pose prior formed by masking the lower half of the target face), a speech encoder for the audio segment, and a face decoder to synthesize the output frame. A key module is a pretrained lip-sync expert discriminator (SyncNet-style), which provides a synchronization loss that penalizes audio–visual mismatch; the generator can also be trained with a separate visual-quality discriminator to improve realism.

Variants of SadTalker [4] and AniPortrait [19]. These variants adapt SadTalker and AniPortrait, originally designed for speech-driven portrait animation, to the lip synchronization setting. The setup matches other lip synchronization models: a driving video provides head pose and other non-lip motion. Since both SadTalker and AniPortrait leverage 3D structure as intermediate features, we replace the non-lip components with those from the driving video and use the predicted lip-related 3D features, resulting in lip synchronization variants.

D.3 Details on the objective metrics

LSE-D and LSE-C. To evaluate lip synchronization accuracy, we measure the audio–visual alignment between lip movements and speech using SyncNet [37], which learns joint representations of audio and visual signals. SyncNet yields two metrics: Lip Sync Error–Distance (LSE-D) and Lip Sync Error–Confidence (LSE-C). LSE-D computes the Euclidean distance between audio and visual embeddings, with lower values indicating better synchronization. LSE-C is a confidence score derived from these distances, where higher values denote more reliable audio–visual alignment.

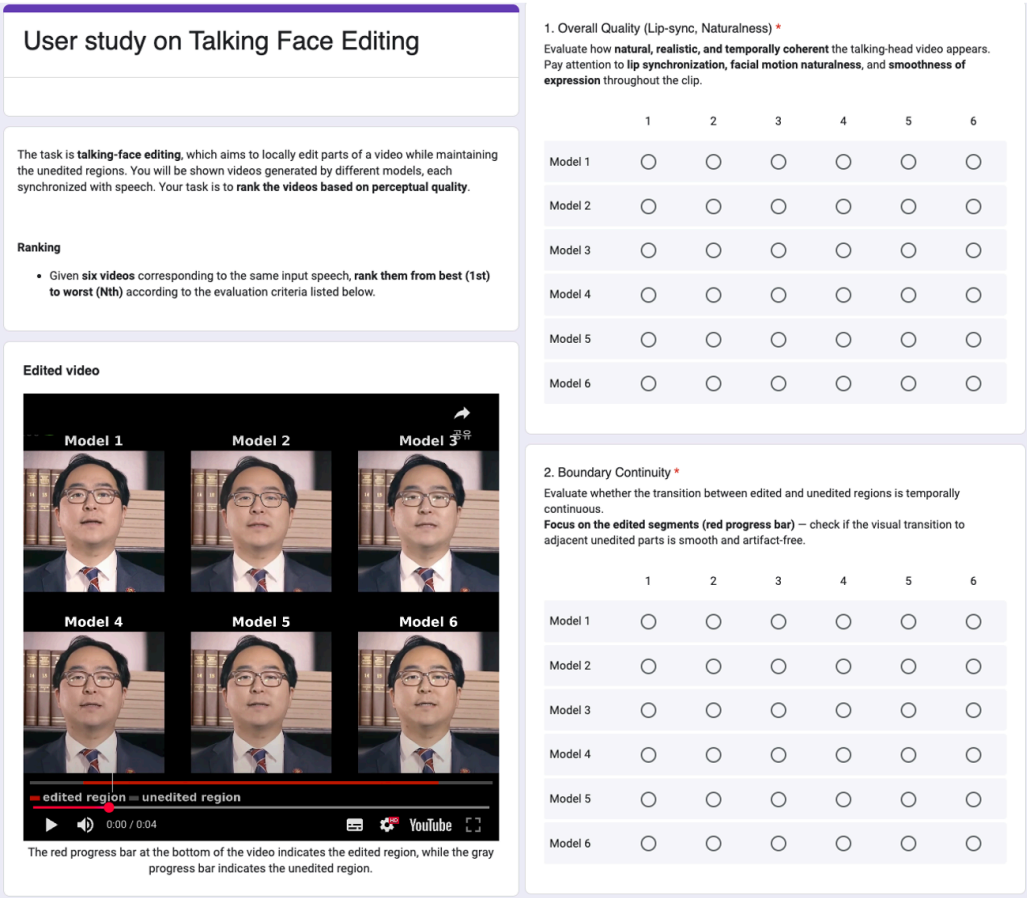


Figure S4: **Instruction and example for the human study.** We provide a sample template used in the talking head video editing evaluation.

Fréchet Video Distance (FVD). To assess perceptual realism in synthesized talking head videos, we employ FVD [38], which measures the distributional similarity between generated and real videos. Following standard practice, we extract video features using a pre-trained I3D network and compute the Fréchet distance between the feature distributions of synthesized and ground-truth videos, capturing both temporal dynamics and overall visual quality.

LPIPS. LPIPS [39] quantifies frame-level perceptual similarity using image features extracted from a pre-trained network. In talking head video synthesis, LPIPS reflects how well identity, facial structure, and appearance are preserved. We compute LPIPS frame-by-frame by comparing matching timestamps from the synthesized and reference videos.

D.4 Human evaluation

We conduct a human study to compare FacEDiT with baselines from speech-driven portrait animation and lip synchronization. We select five recent, high-performing portrait animation models [4, 5, 7, 9, 8] from nine candidates and three lip synchronization models [16, 15, 13]. We form two independent comparison pools, one for each model category, and evaluate 20 synthesized videos per method with 30 participants. For talking head video editing, participants separately rank the methods in each pool by overall quality and boundary continuity. Overall quality considers lip synchronization, facial motion naturalness, and expression smoothness, while boundary continuity measures the temporal smoothness between edited and unedited regions. A red progress bar marks the edit region to help identify the modified segment; the instructions and an example are shown in Fig. S4. We additionally evaluate the corresponding generation task in each pool, *i.e.*, portrait animation or lip synchronization, using overall quality only. Rankings are computed independently within each comparison pool.

E Theoretical interpretation of facial motion infilling

We provide a theoretical interpretation of FacEDiT as a mask-conditional generative model. We first show that multiple talking head synthesis tasks can be expressed as facial motion infilling, and then analyze how random-mask training supports other tasks within a unified model.

Let $\mathbf{F} = (\mathbf{f}_1, \ldots, \mathbf{f}_T) \in \mathbb{R}^{T \times C}$ denote a sequence of facial motion latents, $\mathbf{A}$ denote speech features, and $\mathbf{M} \in \{0,1\}^{T \times C}$ denote a binary spatiotemporal mask, where $\mathbf{M}_{t,j} = 1$ indicates that the corresponding motion entry is masked. The masked and visible motion entries are denoted by $\mathbf{F}_{\mathbf{M}}$ and $\mathbf{F}_{\bar{\mathbf{M}}}$, respectively, *i.e.*, $\mathbf{F}_{\mathbf{M}} = \mathbf{M} \odot \mathbf{F}$ and $\mathbf{F}_{\bar{\mathbf{M}}} = (1 - \mathbf{M}) \odot \mathbf{F}$, where $\odot$ denotes element-wise multiplication. Our facial motion infilling aims to model the conditional probability distribution

$$p(\mathbf{F}_{\mathbf{M}} \mid \mathbf{F}_{\bar{\mathbf{M}}}, \mathbf{A}), \tag{2}$$

i.e., the conditional distribution of missing facial motion given visible motion context and speech.

E.1 Task unification by mask specification

We first show that the probability distribution in Eq. (2) is general, such that it can represent talking head video editing, portrait animation, and lip synchronization as special cases according to the mask.

Proposition 1 (Task unification via motion infilling)**.** *Talking head video editing, speech-driven portrait animation, and speech-driven lip synchronization can all be expressed as facial motion infilling problems by choosing different masks* $\mathbf{M}$*.*

Proof. The proof proceeds by showing that each task can be reformulated as a special case of conditional motion infilling under an appropriate choice of mask $\mathbf{M}$. Specifically, we interpret the mask as specifying which motion components are observed and which components must be generated. Different tasks correspond to different masking strategies, including temporal masking and channel-wise masking.

For talking head video editing, let the edited interval be $[t_1, t_2]$. We define

$$\mathbf{M}_t = \begin{cases} 1, & t_1 \le t \le t_2, \\ 0, & \text{otherwise.} \end{cases}$$

The model predicts motion only inside the edited interval while conditioning on surrounding frames and edited speech. Thus, editing is temporal motion infilling.

For portrait animation, suppose an initial motion prefix up to frame T_0 is observed. We define

$$\mathbf{M}_t = \begin{cases} 0, & t \le T_0, \\ 1, & t > T_0. \end{cases}$$

The model then predicts the remaining motion sequence conditioned on the speech and the observed prefix. Thus, portrait animation is sequence completion via motion infilling.

For lip synchronization, decompose the motion latent as

$$\mathbf{f}_t = \left(\mathbf{f}_t^{\text{rest}}, \mathbf{f}_t^{\text{lip}}, \mathbf{f}_t^{\text{pose}}, \mathbf{f}_t^{\text{trans}} \right).$$

We use the channel-wise mask

$$\mathbf{M}_{t,\text{rest}} = 0, \qquad \mathbf{M}_{t,\text{lip}} = 1, \qquad \mathbf{M}_{t,\text{pose}} = 0, \qquad \mathbf{M}_{t,\text{trans}} = 0.$$

The model reconstructs only lip motion while preserving pose, translation, and non-lip facial motion. Thus, lip synchronization is channel-wise motion infilling.

Therefore, all three tasks are instances of the same conditional generation problem in Eq. (2), with different choices of $\mathbf{M}$. □

This result suggests that a single conditional generative model can perform editing, animation, and lip synchronization within the same probabilistic framework by changing only the masking operator at inference time, *i.e.*, motion infilling as a unified formulation for multiple talking head generation

tasks. In practice, this does not require architectural modification, which is a favorable property of our facial motion infilling formulation.

The above proposition shows that the choice of mask $\mathbf{M}$ defines the task structure through conditional motion infilling, but it does not inform how to obtain the conditional probability distribution. We next demonstrate how our CFM training induces learning of the corresponding conditional probability distribution associated with each mask pattern.

E.2 Mask-conditional flow matching

We first recall the training setup of our method. Let $\mathbf{F}_1 \sim p_{\text{data}}$ be a ground-truth facial motion sequence and $\mathbf{F}_0 \sim p_0$ be a noise sample. Conditional flow matching uses the linear interpolation

$$\mathbf{F}_t = (1-t)\mathbf{F}_0 + t\mathbf{F}_1, \qquad t \sim \mathcal{U}[0,1], \tag{3}$$

whose conditional velocity is

$$\mathbf{u} = \mathbf{F}_1 - \mathbf{F}_0.$$

Given a mask $\mathbf{M}$, FacEDiT conditions the velocity model on speech and visible motion:

$$\mathbf{C}_{\mathbf{M}} = (\mathbf{A}, (1-\mathbf{M}) \odot \mathbf{F}_1, \mathbf{M}).$$

We train the model with the following mask-conditional CFM objective function:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{\mathbf{M}\sim q(\mathbf{M})}\mathbb{E}_{\mathbf{F}_1,\mathbf{A},t,\mathbf{F}_0}\left[\|\mathbf{M} \odot (v_\theta(\mathbf{F}_t, t, \mathbf{C}_{\mathbf{M}}) - (\mathbf{F}_1 - \mathbf{F}_0))\|_2^2\right], \tag{4}$$

where $q(\mathbf{M})$ is the training mask distribution. The model trained with Eq. (4) has the following favorable property.

Proposition 2 (Per-mask optimality under coverage)**.** *Consider random-mask training with $\mathbf{M} \sim q(\mathbf{M})$ using the CFM objective in Eq. (4). Let $\mathcal{M}_{\text{task}}$ be the set of masks required for tasks during inference. Under the assumptions that*

1. *the training mask distribution covers all task masks, $q(\mathbf{M}^\star) > 0, \forall \mathbf{M}^\star \in \mathcal{M}_{\text{task}}$,*
2. *the model class has sufficient capacity to represent the Bayes-optimal conditional velocity fields for every mask condition in $\mathcal{M}_{\text{task}}$,*
3. *optimization reaches the population-risk minimizer,*

the optimal model recovers the Bayes-optimal conditional velocity field over the masked entries for each task mask $\mathbf{M}^\star \in \mathcal{M}_{\text{task}}$.

Its proof is provided below. This proposition suggests that minimizing the CFM objective function learns a mask-conditional velocity field. This velocity field defines a conditional probability flow transporting the prior distribution toward the data distribution under the conditioning $\mathbf{C}_{\mathbf{M}}$. While this objective function does not directly regress the score $\nabla_{\mathbf{F}_{\mathbf{M}}} \log p(\mathbf{F}_{\mathbf{M}} \mid \mathbf{F}_{\bar{\mathbf{M}}}, \mathbf{A})$, velocity fields and score fields are closely related in probability-flow formulations [23]. Therefore, the learned velocity field can be interpreted as implicitly modeling the conditional distribution $p(\mathbf{F}_{\mathbf{M}} \mid \mathbf{F}_{\bar{\mathbf{M}}}, \mathbf{A})$.

According to the proposition, the optimal model is Bayes-optimal for each task mask. This suggests that a single model trained with random masking can cover generation tasks discussed in Proposition 1, including talking head video editing, portrait animation, and lip synchronization, among others.

E.3 Proof of Proposition 2

Before starting the proof, we introduce several definitions and lemmas that are needed to prove the proposition. We first introduce the concept of the Bayes-optimal predictor as follows.

Definition 1 (Bayes-optimal predictor [44])**.** *Let $x \in \mathcal{X}$ denote an input variable, $y \in \mathcal{Y}$ denote a target variable, and $\ell(\hat{y}, y)$ be a loss function. The Bayes-optimal predictor is defined as the function*

$$f^*(x) = \arg\min_{\hat{y}} \mathbb{E}\big[\ell(\hat{y}, y) \mid x\big],$$

which minimizes the conditional expected risk under the true data distribution.

In particular, under squared loss $\ell(\hat{y}, y) = \|\hat{y} - y\|_F^2$, the Bayes-optimal predictor is given by the conditional expectation

$$f^*(x) = \mathbb{E}[y \mid x].$$

We train a generative model with the CFM objective function to approximate the conditional distribution under a mask condition. To connect Bayes optimality with the CFM objective function, we introduce the following lemma.

Lemma 1 (Bayes-optimal conditional velocity)**.** *For a fixed mask* $\mathbf{M}$*, the population minimizer of the masked CFM loss satisfies*

$$\mathbf{M} \odot v^*(\mathbf{F}_t, t, \mathbf{C}_\mathbf{M}) = \mathbb{E}\left[\mathbf{M} \odot (\mathbf{F}_1 - \mathbf{F}_0) \mid \mathbf{F}_t, t, \mathbf{C}_\mathbf{M}\right]. \tag{5}$$

Proof. For fixed $(\mathbf{F}_t, t, \mathbf{C}_\mathbf{M})$, the masked CFM objective is a squared-loss regression problem with prediction $\mathbf{M} \odot v_\theta$ and target $\mathbf{M} \odot (\mathbf{F}_1 - \mathbf{F}_0)$. The squared-loss minimizer is the conditional expectation of this masked target, which gives Eq. (5). □

The above lemma provides a probabilistic interpretation of conditional flow matching under mask-conditioned generation. For each mask condition $\mathbf{M}$, the CFM objective learns the Bayes-optimal conditional velocity field in the least-squares sense.

The learned velocity field defines a conditional probability flow that transports samples from the prior distribution toward the target motion distribution conditioned on $\mathbf{C}_\mathbf{M}$. Consequently, minimizing the CFM objective learns the conditional velocity field associated with the chosen interpolation path under the specified mask.

Although the CFM objective does not explicitly regress the score function $\nabla_{\mathbf{F}_\mathbf{M}} \log p(\mathbf{F}_\mathbf{M} \mid \mathbf{F}_{\bar{\mathbf{M}}}, \mathbf{A})$, probability-flow formulations establish a close relationship between velocity fields and score-based dynamics [23]. Therefore, the learned conditional velocity field can be interpreted as a Bayes-optimal estimator of the conditional generation dynamics under the specified masking condition.

While Lemma 1 characterizes the Bayes-optimal velocity field for a fixed mask condition, the actual training objective optimizes an expectation over masking conditions. Proposition 2 shows that, under a coverage condition on the mask distribution, the resulting predictor remains Bayes-optimal for every condition considered during inference.

Let $\mathcal{M}_{\text{task}}$ be the set of masks required by the downstream tasks. Assume that:

1. the training mask distribution covers all task masks,
$$q(\mathbf{M}^\star) > 0, \qquad \forall \mathbf{M}^\star \in \mathcal{M}_{\text{task}};$$
2. the model class is expressive enough to represent the Bayes-optimal conditional velocity fields for all masks in $\mathcal{M}_{\text{task}}$ simultaneously;
3. optimization reaches the population-risk minimizer.

Then, the optimal model recovers the Bayes-optimal conditional velocity field over the masked entries for each task mask $\mathbf{M}^\star \in \mathcal{M}_{\text{task}}$:

$$\mathbf{M}^\star \odot v_{\theta^*}(\mathbf{F}_t, t, \mathbf{C}_{\mathbf{M}^\star}) = \mathbb{E}\left[\mathbf{M}^\star \odot (\mathbf{F}_1 - \mathbf{F}_0) \mid \mathbf{F}_t, t, \mathbf{C}_{\mathbf{M}^\star}\right]. \tag{6}$$

To prove the above proposition, we use the following lemma for generic per-condition Bayes optimality.

Lemma 2 (Per-condition Bayes optimality under coverage)**.** *Let* $(X, Y, M) \sim p(X, Y, M)$*, where* $M \in \mathcal{M}$ *denotes a condition variable. Define the condition-specific risk*

$$R_M(\theta) = \mathbb{E}_{(X,Y)\sim p(\cdot,\cdot|M)}\big[\ell(f_\theta(X, M), Y)\big].$$

Consider a predictor $f_\theta(X, M)$ *trained to minimize the expected risk*

$$\min_\theta \mathbb{E}_{M\sim p(M)}[R_M(\theta)]. \tag{7}$$

Under the assumptions that

1. *the loss* ℓ *is proper,* e.g.*, squared loss,*
2. *the predictor class has sufficient capacity to represent the Bayes-optimal predictor for every condition in* $\mathcal{M}$*,*

3. *the condition distribution satisfies a coverage condition $p(M) > 0, \forall M \in \mathcal{M}$, where $\mathcal{M}$ includes the set of conditions considered during inference,*

any global minimizer θ^* yields a Bayes-optimal predictor for every condition $M \in \mathcal{M}$:

$$f_{\theta^*}(x, m) = \arg\min_{\hat{y}} \mathbb{E}\big[\ell(\hat{y}, Y) \mid X = x, M = m\big].$$

Moreover, under the squared loss, the Bayes-optimal predictor is

$$f_{\theta^*}(x, m) = \mathbb{E}[Y \mid X = x, M = m].$$

Proof. The expected risk in Eq. (7) can be written as

$$\mathbb{E}_{M \sim p(M)}[R_M(\theta)] = \sum\nolimits_M p(M) R_M(\theta). \tag{8}$$

By the sufficient-capacity assumption, there exists a predictor that simultaneously attains the Bayes-optimal risk for every condition M. Since $p(M) > 0$ for every condition, any global minimizer of the expected risk must therefore attain the minimum conditional risk R_M for every M in the support of $p(M)$. Under a proper loss, minimizing R_M yields the Bayes-optimal predictor for that condition. Furthermore, when the loss is the squared loss, the closed-form Bayes-optimal predictor is given by the conditional mean according to standard regression theory [44]. □

Proof. Proposition 2 is a special case of Lemma 2.

For a fixed mask $\mathbf{M}$, we define the $\mathbf{M}$-conditioned risk

$$R_{\mathbf{M}}(\theta) = \mathbb{E}_{\mathbf{F}_1, \mathbf{A}, t, \mathbf{F}_0}\left[\|\mathbf{M} \odot (v_\theta(\mathbf{F}_t, t, \mathbf{C}_{\mathbf{M}}) - (\mathbf{F}_1 - \mathbf{F}_0))\|_2^2\right]. \tag{9}$$

Then, the CFM objective function in Eq. (4) can be written as

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{\mathbf{M} \sim q(\mathbf{M})}\left[R_{\mathbf{M}}(\theta)\right]. \tag{10}$$

It is clear that Eq. (9) is a special case of Lemma 2. Furthermore, since $R_{\mathbf{M}}(\cdot)$ is the squared loss, the rest of the proof follows Lemma 2.

For each fixed mask, Lemma 1 shows that the Bayes-optimal solution is the corresponding conditional velocity field. Under the realizability assumption, a single model can represent these solutions for all task masks simultaneously. Therefore, the population-risk minimizer recovers the Bayes-optimal conditional velocity field for each covered task mask. In other words, if the training mask distribution sufficiently covers the masking conditions used at inference time, random-mask training allows FacEDiT to support diverse covered task-specific masking patterns within a unified conditional generation framework. This concludes the proof of Proposition 2. □

Remark. Due to the realizability assumption, Proposition 2 should be interpreted as an idealized population-risk analysis of random-mask training. In practice, for a finite-capacity model, a single shared parameterization simultaneously optimizes multiple mask-conditioned risks. As a result, minimizing the expected risk $\mathbb{E}_{\mathbf{M}}[R_{\mathbf{M}}(\theta)]$ may involve trade-offs between different masking conditions, and the learned model may not achieve the Bayes optimum for every mask individually.

Despite this, random-mask training is effective. Under sufficient mask coverage, the training objective encourages the model to learn a shared conditional representation that generalizes across diverse masking conditions. In this sense, FacEDiT can be viewed as amortizing a family of mask-conditioned generation problems associated with $\mathbf{M}$.

F Applications and ethical considerations

FacEDiT offers both valuable applications and important ethical considerations. On the positive side, it offers benefits across domains. For individual creators, online content producers, and professional video production teams, FacEDiT enables precise and efficient editing of localized facial-motion segments without requiring costly or time-consuming reshoots. This capability can significantly improve workflows and reduce production overhead in many real-world applications.

However, as with any generative model, FacEDiT also carries risks of misuse. It could be exploited to generate deepfakes or spread misinformation, raising serious ethical and safety concerns. Responsible deployment is therefore essential, supported by safeguards designed to mitigate these risks. In addition, both video and speech data may encode ethnicity-related biases, which can propagate into the model and result in biased or unfair outputs.

To address these concerns, future work should incorporate techniques, such as robust watermarking and deepfake-detection algorithms specifically designed for facial video content. We also plan to pursue this direction to reduce the risk of FacEDiT being misused. Developing reliable detection and watermarking methods can benefit from access to open-source generative models so that researchers can generate training data and properly evaluate their detectors. Thus, open-sourcing FacEDiT is important for helping the community understand potential misuse scenarios and accelerate research on effective mitigation strategies. Incorporating diverse datasets, including a wide range of ethnicities, facial characteristics, and spoken languages, will further help alleviate demographic biases. Establishing clear usage guidelines and ensuring transparent model governance will support safer and more responsible deployment of the technology.

G Licenses and terms of existing assets

We use existing datasets, pretrained models, codebases, and evaluation tools only for research purposes and cite their original creators. We do not redistribute third-party raw datasets or pretrained checkpoints unless their licenses permit redistribution; any released FacEDiTBench materials will be limited to assets and annotations that comply with the original licenses and usage terms.

- **HDTF**: Released under the CC BY 4.0 license.
- **LivePortrait**: Released under the MIT license. Any optional InsightFace pretrained models used with related pipelines are restricted to non-commercial research use.
- **VoiceCraft**: Code is released under CC BY-NC-SA 4.0, and model weights are released under the Coqui Public Model License 1.0.0.
- **WavLM**: Distributed through Microsoft's UniLM repository, whose root repository is MIT-licensed.
- **RAFT**: Released under the BSD-3-Clause license.
- **MediaPipe**: Released under the Apache-2.0 license.
- **SyncNet**: Released under the MIT license.
- **SadTalker**: Released under the Apache-2.0 license.
- **EchoMimic / EchoMimicV2**: Released under the Apache-2.0 license.
- **Hallo2**: Released under the MIT license, with an S-Lab License 1.0 notice for its high-resolution CodeFormer-derived component.
- **KeyFace**: Released under the Apache-2.0 license.
- **LatentSync**: Code is released under the Apache-2.0 license, with model weights released under OpenRAIL++.
- **Diff2Lip**: Released under the CC BY-NC 4.0 license.

H LLM usage

We used GPT-4o to generate edited transcript candidates for FacEDiTBench, with all prompts and outputs manually inspected. Beyond this use, GPT-4o was used only for grammar and style corrections and was not involved in method design, experimental analysis, claims, or conclusions.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the main contributions, formulating talking head video editing as a facial motion infilling task, introducing FacEDiT, supporting related generation tasks through mask changes, and presenting FacEDiTBench with dedicated metrics. These claims are supported by the method formulation, benchmark construction, and quantitative and qualitative experiments reported in the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6, "Conclusion and discussion" discusses limitations including failure modes inherited from the frozen rendering pipeline, the lack of upper-body motion control, and implicit modeling of emotional context.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important

role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Appendix E, "Theoretical interpretation of facial motion infilling" provides formal theoretical statements and proofs, including task unification by mask specification and mask-conditional flow matching.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 3.5, "Implementation details", Section 5.1, "Experimental setup", Appendix C, "Additional implementation details", and Appendix D, "Details on the experimental setup" provide the model architecture, motion representation, datasets, metrics, baselines, optimizer, training schedule, and inference details needed to reproduce the main results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The project page linked in the abstract provides the training and inference code, curated dataset, and annotations.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 3.5, "Implementation details" specifies architecture, optimizer, learning rate schedule, batch size, training steps, GPUs, temporal smoothness weight, EMA decay, and sampling strategy. Section 5.1, "Experimental setup" describes training/evaluation datasets, competing methods, and objective metrics, while Appendix C, "Additional implementation details" and Appendix D, "Details on the experimental setup" provide further implementation and baseline-adaptation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Although the human evaluation reports variability across participant rankings, the main quantitative benchmark results do not include error bars, confidence intervals, or statistical significance tests.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 3.5, "Implementation details" states that training was performed on two A6000 GPUs with a total batch size of 8 up to 1M steps with early stopping. Appendix B, "Additional results and analysis" reports inference-time comparisons on a single RTX A6000 GPU.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: Appendix F, "Applications and ethical considerations" discusses ethical risks including deepfake misuse, misinformation, and demographic bias, as well as responsible deployment considerations. To the best of our knowledge, the research conforms to the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Appendix F, "Applications and ethical considerations" discusses positive impacts such as reducing costly reshoots and improving workflows for creators and video production teams. The same section also discusses negative impacts, including possible misuse for deepfakes, misinformation, and demographic bias, along with mitigation directions.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [No]

Justification: Appendix F, "Applications and ethical considerations" discusses possible mitigation directions such as robust watermarking, deepfake-detection algorithms, diverse datasets, usage guidelines, and transparent governance. However, the paper does not clearly describe safeguards that have already been put in place for the release of the model or dataset.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: Appendix G, "Licenses and terms of existing assets" lists licenses and usage terms for many datasets, pretrained models, codebases, and evaluation tools used in the paper. However, because some existing assets used in the experiments or dataset construction have unclear or still-unverified license/usage terms, we conservatively answer No.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Section 4, "FacEDiTBench: Benchmark for talking head video editing" introduces FacEDiTBench and documents its composition, edit types, annotations, and evaluation metrics. Appendix A, "Details on FacEDiTBench" further documents the dataset construction pipeline, video sources, transcript generation, edited speech generation, human verification, and metric definitions.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [No]

    Justification: Appendix D.4, "Human evaluation" describes a human study with 30 participants and includes the study instructions and an example screenshot in Figure S4. However, the paper does not provide details about participant compensation.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [No]

    Justification: Appendix D.4, "Human evaluation" reports the human study, but the paper does not describe participant risks, risk disclosure, or whether IRB or equivalent approval was obtained.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Appendix H, "LLM usage" describes our use of GPT-4o for FacEDiTBench transcript generation and writing assistance.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.