

KeyFace: Expressive Audio-Driven Facial Animation for Long Sequences via KeyFrame Interpolation

Supplementary Material

A. Model Details

Implementation In our experiments, both the keyframe generator and the interpolation model produce sequences of 14 frames. The keyframes are spaced by $S = 12$ frames, and the interpolation model uses two frames as conditioning. Consequently, the total number of new frames generated through interpolation is S . This configuration captures extended temporal dependencies while maintaining computational efficiency.

We initialize the weights of the U-Net and VAE from SVD [3] and conduct all experiments on NVIDIA A100 GPUs with a batch size of 32 for both models. The keyframe generator is trained for 60,000 steps, while the interpolation model requires 120,000 steps due to its greater deviation from the pre-trained SVD. We use the AdamW optimizer [40] with a constant learning rate of 1×10^{-5} , following a 1,000-step linear warm-up. For inference, we use 10 steps, consistent with [4]. During training, the identity frame is randomly selected from each video clip.

Audio is sampled at 16,000 Hz to align with the pre-trained encoders (WavLM [8] and BEATs [9]), while video frames are extracted at 25 fps and resized to 512×512 pixels. During training, the audio condition is randomly dropped 20% of the time, and the identity condition is dropped 10% of the time to strengthen the guidance effect.

We train the reduced model for autoguidance [32] with $16\times$ fewer training steps. The default settings are summarized in Table 8.

Parameter	Value
Keyframe sequence length (T)	14
Keyframe spacing (S)	12
Interpolation sequence length (S)	12
Keyframe training steps	60,000
Interpolation training steps	120,000
Training batch size	32
Optimizer	AdamW
Learning rate	1×10^{-5}
Warm-up steps	1,000
Inference steps	10
GPU used	NVIDIA A100
Autoguidance [32] model training steps	$120,000 / 16 = 7,500$
Audio condition drop rate for CFG [21]	20%
Identity condition drop rate for CFG [21]	10%
Audio CFG [21] scale	3
ID CFG [21] scale	2.33

Table 8. Default model parameters and training configurations.

Inference speed One limitation of our model is that it does not yet support real-time generation. Nevertheless, our two-stage approach is faster than competing diffusion-based models, particularly because it allows batching, unlike autoregressive methods. We present an inference speed comparison (Table 9), measured in seconds per frame. Real-time inference could potentially be achieved through distillation methods (e.g., UFOGen), which we leave for future work.

V-Express [59]	Hallo [65]	AniPortrait [63]	EchoMimic [11]	Keyface
3.36	1.9	0.44	0.76	0.26

Table 9. Seconds per frame comparison for baseline models.

B. Comparison with SVD

Our method builds upon Stable Video Diffusion (SVD) [3] by introducing carefully designed architectural and task-specific adaptations. These modifications distinctly set our approach apart from prior work. We highlight the primary differences below.

Audio Conditioning While SVD primarily conditions on the initial frame to predict subsequent video frames, our method extends this capability by conditioning on both an identity frame and audio inputs to drive video generation. To the best of our knowledge, we are the first to employ conditioning based on outputs from two distinct audio encoders (WavLM [8] and BEATs [9]), allowing simultaneous processing of speech and non-speech audio.

Emotional Conditioning Unlike the original SVD architecture, our approach incorporates additional control over emotional expression. We demonstrate that training emotional models exclusively with pseudo-labels for valence and arousal achieves robust and consistent performance.

Loss Functions SVD employs only the EDM loss [31]. In contrast, we use two additional pixel-space losses along with a weighted loss that specifically targets the lower region of generated images.

Guidance Whereas SVD solely employs vanilla classifier-free guidance (CFG) [21], we provide an in-depth investigation into optimal guidance techniques

tailored specifically to each stage of our pipeline. We found that, for the keyframe model, assigning different CFG weights to identity and audio conditions leads to better performance and improved robustness compared to classical CFG. Additionally, since interpolation requires greater flexibility in head movement, we employed autoguidance [32] to dynamically balance guidance, resulting in enhanced overall video quality.

C. Datasets

C.1. Data details

Table 10 provides an overview of the datasets used in this paper, detailing the number of speakers, videos, average video duration, and total duration for each dataset. We use a combination of publicly available datasets (HDTF [76], CelebV-HQ [78], CelebV-Text [70]) and our own collected data. As stated in the main paper, we use only HDTF and the collected data for training our final model. Additionally, we utilize reference frames from FEED [14] for some qualitative results.

Dataset	# Speakers	# Videos	Duration	
			Avg. (sec.)	Total (hrs.)
HDTF [76]	264	318	139.08	12
CelebV-HQ [78]	3,668	12,000	4.00	13
CelebV-Text [70]	9,109	75,307	6.38	130
Collected data	824	4,677	123.15	160
Collected data (NSV)	639	5,701	18.94	30

Table 10. Overview of the datasets used in the study.

C.2. Preprocessing details

Even during our experimentation with alternative data sources in the data ablation study, we aim to obtain the highest-quality data possible. To achieve this, we propose a data preprocessing pipeline with the following steps:

- Extract 25 fps video and 16 kHz mono audio.
- Discard low-quality videos based on a quality score computed using HyperIQA [53].
- Detect and separate scenes using [PySceneDetect](#).
- Remove clips without active speakers using [Light-ASD](#) [39].
- Estimate landmarks and poses using [face-alignment](#).
- Crop the video around the facial region across all frames.

Using this pipeline, we curate CelebV-HQ [78] and CelebV-Text [70].

However, even after filtering the datasets, we found that many samples contain editing effects and/or occlusions that are not detected. Examples include visible hands, camera movement, editing effects, and occlusions, which we found occur in 20 % of videos even after our cleaning process, as illustrated in Figure 9. Since these artefacts don’t correlate

with speech, they can’t be replicated by the model, hindering performance as shown in Section 5.3.



Figure 9. Illustration of bad examples in CelebV-HQ [78] and CelebV-Text [70].

D. Evaluation metrics

D.1. LipScore

To evaluate the effectiveness of our proposed LipScore metric compared to the traditional SyncNet metric, we conduct experiments introducing controlled temporal and spatial perturbations to synchronized audio-visual data. The goal is to observe how each metric responds to these perturbations and determine which better correlates with the expected degradation in lip synchronization quality.

Temporal misalignment sensitivity In the first set of experiments, we introduce temporal misalignments by shifting the ground truth video temporally. The time shifts range from 0 milliseconds (ms) to 1000 ms.

Figure 10 illustrates the behavior of SyncNet Confidence and SyncNet Distance as functions of the time shift. We observe that SyncNet Confidence and Distance remain constant up to approximately 400 ms and only start to change significantly beyond this point. This behavior is undesirable, as even small misalignments (e.g., 100–200 ms) should result in a noticeable decrease in confidence and an increase in distance.

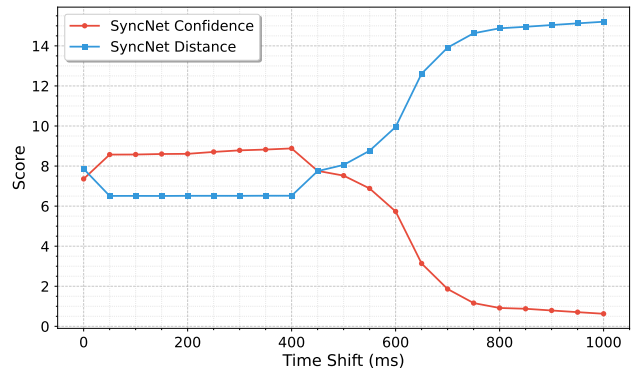


Figure 10. SyncNet Confidence and SyncNet Distance as functions of time shift (ms).

In contrast, Figure 11 shows the LipScore metric’s response to the same range of time shifts. LipScore exhibits a

stable and consistent decrease in score as the time shift increases. It begins to penalize even small temporal perturbations, with a sharp decline at smaller offsets, and stabilizes at lower scores as larger misalignments are introduced. This behavior aligns with the expected characteristics of a robust lip synchronization metric, demonstrating continuous sensitivity to temporal misalignments without erratic or overly abrupt changes.

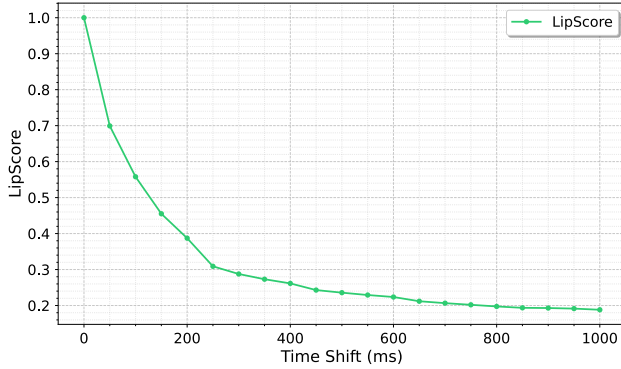


Figure 11. LipScore as a function of time shift (ms).

Robustness to spatial perturbations We evaluate the robustness of the metrics to spatial transformations by introducing horizontal shifts and rotations to the video frames.

Figure 12 illustrates the percentage deviation from the initial metric values as horizontal shifts increase. LipScore remains stable, exhibiting minimal deviation across the range of horizontal shifts, indicating its robustness to this type of spatial perturbation. In contrast, SyncNet Confidence and SyncNet Distance show significant deviations starting at a shift of 75 pixels, highlighting their sensitivity to horizontal displacements.

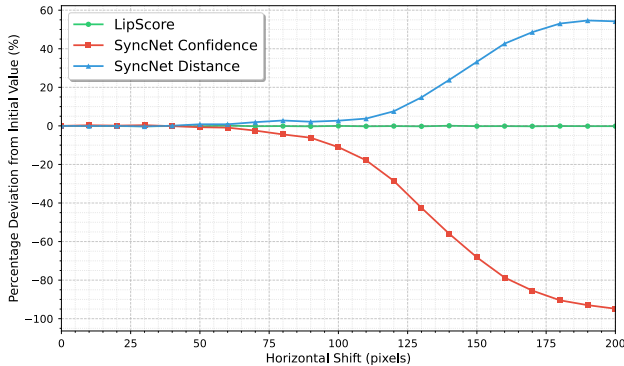


Figure 12. Effect of horizontal shifts on LipScore, SyncNet Confidence, and SyncNet Distance. The plot shows the percentage deviation from the initial value as the horizontal shift increases.

Similarly, Figure 13 shows the percentage deviation in

metric values as the rotation angle of the video frames increases. LipScore again demonstrates robustness, with negligible changes in its values even as the rotation angle grows. In contrast, SyncNet Confidence and SyncNet Distance exhibit substantial deviations starting at 20 degrees, indicating that these metrics are more adversely affected by rotational transformations.

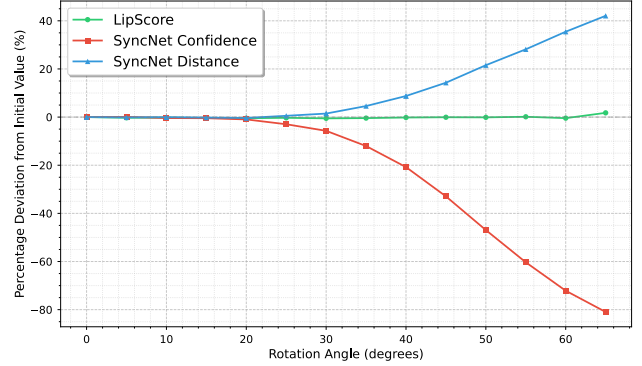


Figure 13. Effect of rotation angles on LipScore, SyncNet Confidence, and SyncNet Distance. The plot shows the percentage deviation from the initial value as the rotation angle increases.

WER on unseen datasets We additionally evaluate our state-of-the-art lipreader [41] on HDTF and find that it achieves a 21 % WER, demonstrating strong performance on unseen data and further supporting LipScore’s validity.

D.2. Non-speech vocalization classifier

We introduce the Non-Speech Vocalization (NSV) Classifier as part of our evaluation methodology. This not only highlights the limitations of pre-trained speech-driven animation methods but also demonstrates the capabilities of our model in generating realistic NSV sequences. The model processes video inputs and classifies them into one of eight NSV types, plus speech.

Architecture The architecture of the system is presented in Fig. 14. We employ a Multiscale Vision Transformer (MViTv2) [37] backbone, augmented with two linear layers and a dropout layer with a dropout probability set to 0.2. The MViTv2 model, pre-trained on the Kinetics dataset [33], achieves a top-5 accuracy of 94.7 %.

Training Our model is trained using a dataset containing video clips of eight different NSV types and speech. The eight NSV classes are: “Mhm”, “Oh”, “Ah”, *coughs*, *sighs*, *yawns*, *throat clears*, and *laughter*. During the training process, video clips corresponding to any of these classes are fed into the model. We train using the AdamW

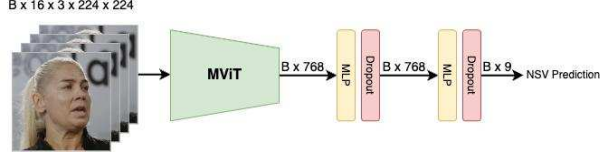


Figure 14. The architecture used for the Non-Speech Vocalization Classifier. The batch size is denoted as B.

optimizer with a learning rate of 1×10^{-4} , $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The cross-entropy loss is employed as the loss function.

Our model achieves an F1 score of 0.7 across these nine classes, demonstrating its effectiveness in classifying various NSVs and speech.

NSVs performance boundaries To demonstrate and understand the effectiveness of NSV_{acc} across individual NSVs, we present a confusion matrix on the validation set of the data used to train NSV_{acc} (Fig.15, left). Although the model achieves good overall performance, certain NSVs are frequently confused, such as “Oh” with “Ah,” “Sigh” with “Mhm,” and “Yawn” with “Cough.”

Additionally, we demonstrate that our model can generate visually distinct NSVs (Fig.15, right) with few confusions by generating 10 videos per NSV category and speech.

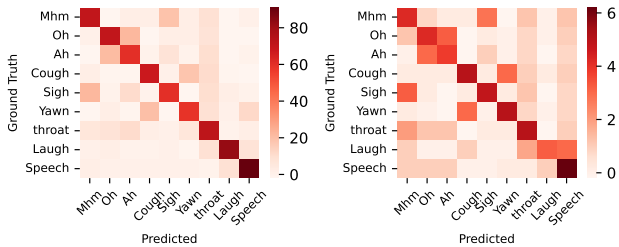


Figure 15. NSV confusion matrix for generated (left) and validation (right) videos.

E. User study details

To evaluate the performance of our proposed method, KeyFace, against existing baselines, we conduct a comprehensive

user study. Participants view pairs of talking face videos and select the one they find more realistic. This section summarizes the results of the pairwise comparisons and the derived metrics.

Pairwise Win Rates: The pairwise win rate matrix is presented in Figure 16. Each cell represents the proportion of times the reference model (rows) is preferred over the competing model (columns). Green indicates a high win rate for the reference model, while red represents a lower win rate. KeyFace is consistently preferred over baseline models, achieving a win rate of at least 64 % against all other methods.

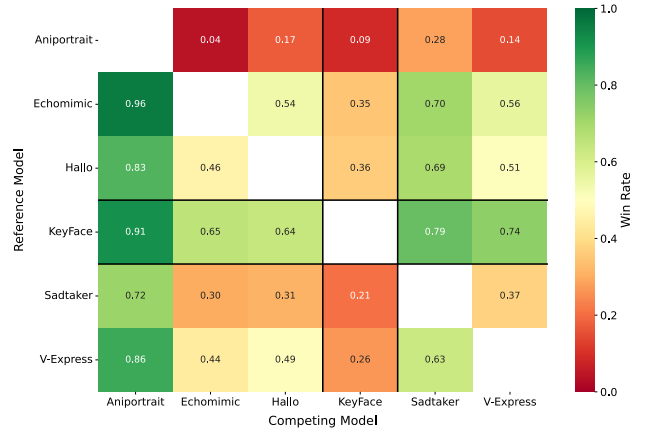


Figure 16. Pairwise win rates between reference (rows) and competing models (columns). Green indicates higher, Red lower win rates.

Elo ratings: Figure 17 presents the Elo ratings for all models with 95 % confidence intervals. KeyFace achieves the highest Elo rating, significantly outperforming the baselines, demonstrating its effectiveness in generating high-quality talking face animations.

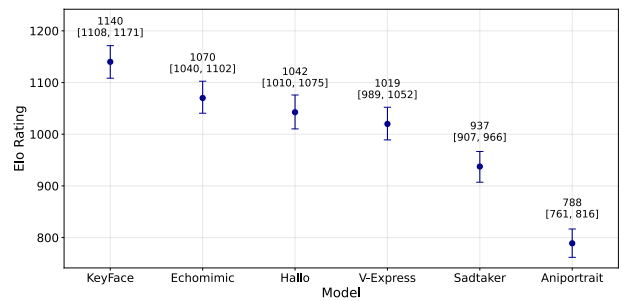


Figure 17. Elo ratings for all models with 95 % confidence intervals. Higher ratings indicate better overall performance.

Elo rating distributions: The density distributions of Elo ratings are shown in Figure 18. KeyFace exhibits a sharp, high-density peak at the upper end, highlighting its robustness and consistent user preference across evaluation scenarios. Echomimic, V-Express, and Hallo show significant overlap in their results, while Aniportrait and SadTalker consistently receive lower ratings.

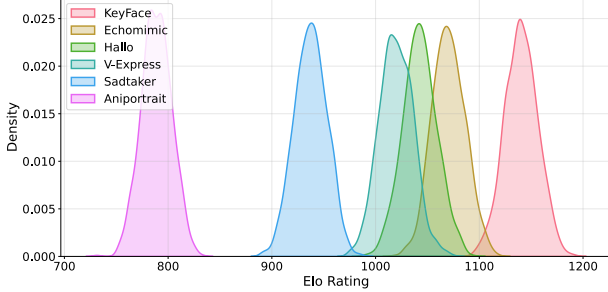


Figure 18. **Density distributions of Elo ratings for all models.** Peaks indicate the most probable performance levels, with higher ratings reflecting better performance.

F. Additional ablation

Method	FID ↓	FVD ↓	LipScore ↑
w/o cross attention	16.95	167.39	0.35
w/o timestep	17.20	176.83	0.28
cross attention + timestep	16.76	137.25	0.36

Table 11. **Audio conditioning ablation** on HDTF [76]: “Cross attention” refers to incorporating audio through a cross-attention mechanism, while “timestep” refers to adding the audio embeddings to the timestep embeddings. The best results are highlighted in **bold** and default settings are highlighted in **gray** on all tables.

Audio mechanisms Table 11 presents an ablation study on the impact of different audio conditioning mechanisms on video generation quality. The results show that the audio timestep plays a critical role in achieving accurate lip synchronization, as removing it (row “w/o timestep”) results in the lowest LipScore and the highest FVD. Adding cross attention alone improves video quality but only marginally enhances the LipScore compared to when the timestep is absent. The best performance is achieved when both cross attention and audio timestep embeddings are used together, leading to the lowest FID, significantly lower FVD, and the highest LipScore. This indicates that while audio timestep embeddings are essential for achieving good lip synchronization, the addition of cross attention further enhances the overall quality of the generated videos by improving visual coherence and temporal consistency.

Training on HDTF only To ensure a fair comparison with baseline models, we retrain our model exclusively on publicly available data (i.e. HDTF [76]), removing all non-public sources. Although this leads to a decrease in performance, our model still outperforms baseline methods trained on larger datasets. We emphasize that most existing methods rely on private datasets; therefore, to maintain fairness, we curated our dataset to have comparable scale in terms of total hours and number of speakers as described in Section C.1.

Method	FID ↓	FVD ↓	LipScore ↑
KeyFace (HDTF only)	19.49	165.06	0.28

Table 12. Results of pipeline trained on HDTF only.

G. Limitations

One key limitation of our model, which it shares with all baseline methods, is its performance when the initial frame exhibits an extreme head pose. This issue primarily stems from the lack of training data containing such extreme poses, resulting in difficulties in reconstructing the occluded or unseen parts of the face. As illustrated in Figure 19, although the model can generate plausible videos with accurate lip synchronization, it partially loses the identity of the reference image in these scenarios. Additional failure cases involving challenging reference frames are provided in the supplementary videos.

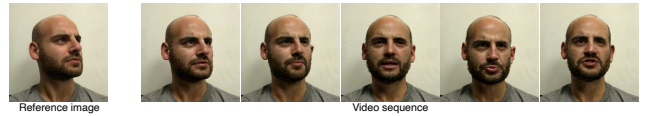


Figure 19. An example showcasing KeyFace’s limitations in handling extreme head poses.

H. Additional qualitative results

To further demonstrate the effectiveness of our method, we provide **example videos generated by KeyFace** (as well as competing methods, for comparison) in the supplementary material:

- **Non-speech vocalizations comparison.** We evaluate the model’s ability to handle eight distinct NSVs and compare its performance with baseline methods, highlighting the limitations of current state-of-the-art models and the strengths of our approach. For a fair comparison, all examples maintain a neutral emotional tone.
- **Speech and NSV comparison.** We demonstrate the model’s capability to generate both speech and NSVs

within the same video, comparing its performance to other approaches. The results showcase the holistic nature of our method, particularly in contrast to baseline models. We maintain a neutral emotional tone for consistency.

- **Side-by-side comparison.** We present side-by-side comparisons between KeyFace and baseline models, showcasing KeyFace’s superior performance in generating realistic and expressive facial animations.
- **Emotion interpolation.** We showcase transitions between different emotional states, emphasizing the model’s ability to capture subtle and nuanced expressions.
- **Out-of-distribution robustness.** Figure 20 illustrates the model’s robustness in handling non-human faces, demonstrating successful generalization to a variety of input conditions.
- **Expanded KeyFace examples.** We provide additional videos featuring KeyFace-generated animations in English and other languages, highlighting the model’s generalization capabilities across different linguistic contexts.



Figure 20. We present a set of examples with **out-of-distribution** reference frames.