

Article

GANimate: Ultra-Efficient Lip-Landmark-Driven Talking Face Animation Using a Learned Kalman Filter on GAN Feature Latent Space for Human–Computer Interaction on Mobile Devices

Ethan Fenakel *, Ben Ohayon and Dan Raviv

School of Electrical and Computer Engineering, Tel Aviv University, Tel Aviv 6997801, Israel

* Correspondence: ethanfenakel@mail.tau.ac.il

Abstract

We present GANimate, a lightweight method for animating talking faces that leverages recent advances in latent-space manipulation of Generative Adversarial Networks (GANs). Unlike existing approaches based on computationally intensive diffusion models, transformers, or complex 3DMM representations, which are impractical for mobile and other low-resource edge devices due to high memory and compute demands, GANimate is designed for efficient operation on low-memory, low-compute edge devices. The model operates on 2D lip landmarks extracted from standard mobile vision-sensor inputs and requires no pre-training, making it easily integrable with any lip-landmark generator. Through an optimization process in the GAN feature latent space, these landmarks act as geometric constraints to animate a static portrait, producing realistic and expressive lip movements. To maintain stability and visual coherence across frames, we employ a Kalman filter to detect and track lip landmarks during video synthesis, enabling adaptive refinement and improved temporal consistency. The result is a compact and modular framework that bridges the gap between performance and accessibility in talking face synthesis, delivering high-quality and stable animations with minimal computational overhead. GANimate represents an important step toward lifelike, real-time avatars suitable for sensor-enabled and mobile human–computer interaction.

Keywords: video generation; animation; interactive image manipulation; facial animation; GANs



Academic Editor: Cosimo Distanto

Received: 3 January 2026

Revised: 4 February 2026

Accepted: 18 February 2026

Published: 22 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Facial expressions are a powerful means of communication, conveying emotion and enriching interaction. With the widespread availability of vision sensors on mobile and edge devices, such as smartphone cameras, tablets, and embedded systems, facial animation and talking face avatars have become increasingly relevant for sensor-enabled human–computer interaction (HCI). Advances in generative AI and deep learning have enabled talking face avatars, enhancing human–AI and human–human interaction in applications including remote tutoring, mental health support, virtual assistants, teleconferencing, and gaming.

Talking face synthesis aims to generate realistic facial motion from a static portrait image, such as one captured by a mobile phone camera, driven by input signals including video, audio, or, in our case, 2D lip landmarks extracted from audio captured by mobile devices. While recent methods [1–7] have improved realism and quality, many remain

impractical for mobile and low-resource edge platforms due to memory-intensive complex architectures, high latency, or reliance on pre-training for real-world generalization, which limits their applicability in HCI scenarios.

Some methods [7–9] use 2D or 3D facial landmarks as intermediate representations to guide expression transfer in talking face synthesis. While these geometric inputs are interpretable, they struggle to preserve fine details and typically fail to generate high-quality videos. Alternatively, other approaches [10–14] rely on 3D Morphable Models (3DMMs), which decompose facial states into expression, pose, and identity, enabling more granular control. Although compact, 3DMMs fall short in capturing detailed expressions. In audio-driven settings, audio features are often projected into the 3DMM parameter space [12,15,16], allowing realistic synchronization of lip motion and expressions.

Generation efficiency is crucial for HCI applications on mobile devices. Although diffusion techniques have greatly improved visual fidelity in talking face synthesis [4,17–19] and general video generation [20,21], their high computational cost, large memory footprint, and long inference times severely limit their usability on mobile and embedded platforms. There is a clear need for lightweight, optimized algorithms that balance quality with low-latency performance.

Recently, image editing techniques have made notable progress by using the powerful pre-trained StyleGAN2 [22], a popular Generative Adversarial Network (GAN) consisting of a generator and a discriminator trained in an adversarial manner to synthesize realistic images. StyleGAN2-based editing has achieved strong results in facial attribute editing, pose changes, and stylization. Inversion techniques [23–27] enable high-quality face manipulation by leveraging StyleGAN’s image prior, removing the need for training large models while preserving 1024×1024 resolution and fine details.

Building on the success of GAN-based editing methods like DragGAN [28] and FreeDrag [29], which enable precise point-based image manipulation via latent space optimization, we introduce GANimate, a novel approach for realistic talking face generation tailored to mobile settings.

GANimate takes a static face image, performs PTI inversion [27] to obtain its latent code, and combines it with lip landmarks to drive animation through lip geometry-constrained optimization in StyleGAN2’s latent space [22].

While existing point-based editing methods often suffer from temporal instability, jitter, and visual artifacts when applied to video, we address this challenge by integrating a Kalman filter [30] to robustly track and smooth lip landmarks over time. This tracking mechanism improves temporal coherence, reduces visual artifacts, and supports the low-latency and low computational resource requirements of edge devices. When paired with a lip motion generator such as Wav2Lip [31], GANimate supports future real-time use, enhancing interactive human–AI experiences.

Our key contributions can be summarized as follows:

- (i) A highly lightweight talking face animation framework (56M parameters at 1024 resolution, 30M at lower resolutions) [22], orders of magnitude smaller than diffusion-based models (860M parameters) [32], enabling future deployment on mobile and edge devices;
- (ii) A modular, separable design that integrates seamlessly with any lip-landmark generator without requiring end-to-end training or pre-training;
- (iii) An adaptive inference time refinement mechanism that continuously improves animation quality during generation.

A demonstration of GANimate is included in the Supplementary Materials, illustrating the generated talking face animations.

2. Related Work

2.1. Talking Face Generation

Talking face synthesis has evolved from identity-specific image-to-image translation models [33–35] to few-shot meta-learning [8,36] and more recently to subject-agnostic methods using a single image [9,37–40]. For example, Monkey-Net [41] mapped sparse to dense motion flows, while FOMM [9] introduced first-order affine transformations for finer motion transfer. Subsequently, Face-vid2vid [40] advanced this by using unsupervised 3D keypoint detection to support free-view talking head generation.

Other 3D-based methods leverage models like 3DMM [42] for more controllable animation. DVP [43], NS-PVD [44], HeadGAN [10], and PIRenderer [11] synthesize subject-agnostic portraits using parametric control, but often lack fine details.

Another line of work focuses on audio-driven approaches translating speech into facial motion. Early works like Synthesizing Obama [45], NVP [46], and AudioDVP [47] focused on speaker-specific models. Generalizable methods include Speech2Vid [48], which introduced end-to-end lip syncing, and its adversarial improvement [49]. Wav2Lip [31] used inpainting to synchronize lips with audio, while PC-AVS [50] ensured seamless pose and lip alignment. Transformer-based models like [40] enhanced accuracy using pre-trained FOMM [9]. To improve realism and generalization, many methods use structural cues like facial landmarks [7,51,52], motion flows [13], or 3D meshes [53]. StyleHEAT [14] combines StyleGAN2 [22] with 3DMM to generate high-res animations with detailed expressions, poses, and identities, but its reliance on 3DMM and a separate motion generator increases complexity.

2.2. Image Editing via Pre-Trained StyleGAN

Many methods edit images by manipulating GAN latent vectors. Some approaches derive meaningful latent directions through supervised learning, utilizing manual annotations or prior 3D models [23,24,54–56]. Others identify important semantic directions in the latent space using unsupervised techniques [57–59]. Recent methods like GANWarping [60] and UserControllableLT [61] support point-based editing but face limitations: GANWarping handles out-of-distribution edits without ensuring realism, while UserControllableLT allows only single-point edits and struggles with multi-point accuracy. DragGAN [28] has gained attention for its strong performance, combining point tracking with motion supervision on feature maps to move input points accurately toward targets. It inspired follow-ups like FreeDRAG [29], which introduces adaptive template features for more flexible updates and improved resilience to tracking failures.

3. Method

3.1. Problem Setting and Initialization

Our goal is to synthesize a talking face video driven by an online stream of lip landmarks. The input is a single face image $I_o \in \mathbb{R}^{3 \times H \times W}$ of an arbitrary person and a sequence of facial landmarks $\{T_t\}_{t=0}^T, T_t \in \mathbb{R}^{68 \times 2}$. The facial landmarks are assumed to be provided by an external landmark generation module (e.g., an audio-driven landmark prediction model such as [62] or detected from a video using FaceAlignment [63]) and do not need to be extracted from the target person's video. Since target landmarks may originate from external sources with different coordinate systems (e.g., varying image resolutions or pose differences), we align the target landmarks to the coordinate system of the first frame using the Umeyama algorithm [64], which estimates a similarity transformation between the two landmark sets. The output is a video $\{I_t\}_{t=0}^T, I_t \in \mathbb{R}^{3 \times H \times W}$, where in each frame t , the lips are synchronized with T_t , maintaining realism and stability.

We build on StyleGAN2 [22], where a 512-dimensional latent code $z \sim \mathcal{N}(0, I)$ is mapped to an intermediate latent code $w \in \mathbb{R}^{512}$, which is then injected into the generator to synthesize an image $I = G(w)$. To animate arbitrary faces, we apply PTI [27] to invert the input image I_o into the latent space, optimizing both w and G to faithfully reconstruct the face and preserve identity, producing an initial latent code for animation.

3.2. Landmark-Based Face Manipulation

Figure 1 illustrates our landmark-based image manipulation. Given a GAN-generated face image $I_t \in \mathbb{R}^{3 \times H \times W}$, we detect 68 facial landmarks using FaceAlignment [63] and extract the 20 lip landmarks (indices 48–68) as the frame’s handle points. The same indices are extracted from the input landmark stream T_t to serve as target points. We define them as:

$$\text{Handles: } \{p_i^t = (x_{p,i}^t, y_{p,i}^t) \mid i = 1, \dots, 20\} \quad \text{Targets: } \{t_i^t = (x_{t,i}^t, y_{t,i}^t) \mid i = 1, \dots, 20\} \quad (1)$$

For each frame t , the goal is to align each lip landmark p_i^t with its target t_i^t to produce natural talking animation. This is done by optimizing the image’s latent features based on lip landmarks, using two steps per frame: motion supervision and point tracking, as in [28].

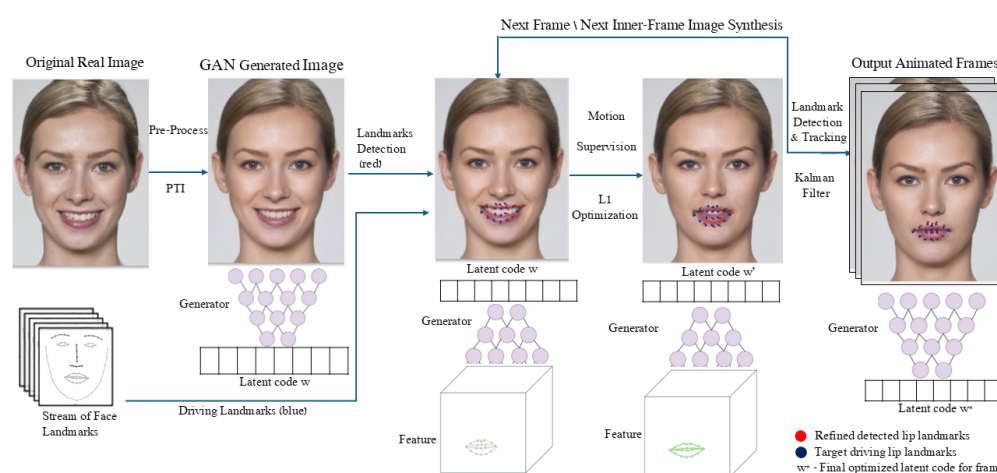


Figure 1. Framework overview. Given a GAN-generated or inverted image and a set of driving lip landmarks, our method iteratively applies motion supervision (Section 3.3) and landmark tracking (Sections 3.4 and 3.5). Motion supervision aligns detected landmarks with the target landmarks to generate animation, while tracking ensures smooth and stable transitions within and across frames.

In the motion supervision stage, an L_1 loss encourages handle points to move toward target points by matching feature-space neighborhoods. This optimizes the latent code w , yielding an updated w' and image I' , with lip landmarks slightly shifted toward targets. As this step does not ensure precise alignment, lip detection and tracking are needed to update handle points. Detection and tracking serve two roles: correcting landmark inaccuracies to prevent distortions and ensuring smooth transitions for natural lip motion and stable, realistic video. Each frame undergoes up to 10 iterations or stops once handle landmarks align with targets, which on average takes 5 iterations, producing a video frame.

3.3. Motion Supervision

Following DragGAN [28], we utilize the discriminative power of the generator’s intermediate features, enforcing an L_1 loss between neighboring features to guide the latent motion of each landmark from handle to target. Specifically, we use the feature map $F \in \mathbb{R}^{128 \times 256 \times 256}$ from the sixth block of the StyleGAN2 [22] generator, which offers the

best balance between discriminability and resolution, as described in [28]. To align with the frame size, F is resized using bilinear interpolation from its 256×256 resolution.

A small patch r around each handle point p_i^t guides its movement toward the target point t_i^t . To preserve facial identity and expression, we use a binary mask $M_t \in \{0, 1\}^{H \times W}$ that restricts motion supervision to specific pixel locations while leaving other regions of the feature map F untouched. The supervised area in M_t is defined by a convex hull around the lip landmarks.

The motion supervision loss we use is:

$$L = \sum_{i=1}^{20} \sum_{q \in \Omega(p_i^t, r)} \|F(q_i^t) - F(q_i^t + d_i^t)\|_1 + \lambda \| (F - F_0)(1 - M_t) \|_1 \quad (2)$$

where $\Omega(p_i^t, r)$ denotes the pixels within radius r of landmark p_i^t , $F(q_i^t)$ are feature values at q_i^t , and $d_i^t = \frac{t_i^t - p_i^t}{\|t_i^t - p_i^t\|_2}$ is the unit vector from the target to handle landmark. The first term sums up all landmarks to supervise the motion of the entire lips. During backpropagation, gradients are blocked through $F(q_i^t)$, ensuring that p_i^t moves towards $p_i^t + d_i^t$ and not vice versa. The second term, weighted by λ , enforces a reconstruction constraint on unmasked regions defined by the binary mask M_t , preventing unintended feature changes outside the lip motion area, balancing motion and stability. In each motion supervision stage, this loss optimizes the latent code w for one step, as F is a function of w , modifying only the first six layers to adjust spatial attributes while keeping the rest frozen to preserve appearance. Each motion supervision step gradually shifts the lip landmarks toward their targets in the next frame.

3.4. Point Tracking

After motion supervision, the updated latent code w' , feature map F' , and image I' are obtained. However, the new handle landmark positions are unknown, requiring an update before the next optimization step. Previous works introduced traditional tracking methods including optical flow [65–69], nearest-neighbor search [28], and template feature updates [29], which often compromise efficiency or introduce accumulative errors. In particular, the nearest-neighbor patch search [28] is unsuitable for lip-landmark tracking due to the uniform texture in the lip region, making it difficult to extract precise landmark updates from GAN features. This destabilizes lip motion and introduces artifacts in the video.

To address this, we replace GAN feature-based tracking with an accurate and efficient face landmark detector [63]. Specifically, at each point tracking stage, FaceAlignment is applied to the updated image I' to directly determine the current landmark locations. The detected landmarks are then used as updated handle positions for the next iteration of the optimization process. Unlike prior feature-space tracking approaches [28,29], our method updates landmarks by detecting them on the generated image, which improves robustness for lip-landmark tracking in low-texture regions. However, since landmark detectors may introduce small frame-to-frame variations, minor detection noise can introduce local errors, causing jitter and instability in lip motion, which may affect subsequent motion supervision and degrade video quality. To counteract this and enhance motion smoothness, we apply a Kalman filter [30] during frame generation and interframe transitions. Each of the 20 lip landmarks is individually tracked using a constant acceleration motion model, ensuring precise motion prediction and improving realism in the generated video.

3.5. The Discrete Kalman Filter

The Kalman filter is a standard tool for real-time signal tracking in control systems and computer vision. It estimates the optimal state in a linear dynamic system by combining predictions and noisy observations using a weighted average based on state covariance. The filter estimates the state s_i^t of a discrete-time process using:

$$s_i^t = As_i^{t-1} + u_i^t \quad (3)$$

$$p_i^t = Hs_i^t + v_i^t \quad (4)$$

Here, A and H are the transition and observation matrices; $u_i^t \sim \mathcal{N}(0, Q)$ and $v_i^t \sim \mathcal{N}(0, R)$ are independent process and measurement noise terms. The measurement p_i^t and the state vector s_i^t are given by:

$$p_i^t = (x_{p,i}^t, y_{p,i}^t) \quad (5)$$

$$s_i^t = (x_{p,i}^t, y_{p,i}^t, Vx_{p,i}^t, Vy_{p,i}^t, ax_{p,i}^t, ay_{p,i}^t) \quad (6)$$

where $(Vx_{p,i}^t, Vy_{p,i}^t)$ are the point velocities and $(ax_{p,i}^t, ay_{p,i}^t)$ are point accelerations.

Following the constant-acceleration motion model described in [70], we define the state transition matrix A to predict the landmark state at the next iteration given the input landmark's frame rate dt . The observation matrix H extracts the landmark position from the state vector, as defined in Equations (4) and (6):

$$A = \begin{pmatrix} 1 & 0 & dt & 0 & \frac{1}{2}dt^2 & 0 \\ 0 & 1 & 0 & dt & 0 & \frac{1}{2}dt^2 \\ 0 & 0 & 1 & 0 & dt & 0 \\ 0 & 0 & 0 & 1 & 0 & dt \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (7)$$

Process and measurement noise covariances are as follows [70]:

$$Q = \begin{pmatrix} \frac{1}{4}dt^4 & 0 & \frac{1}{2}dt^3 & 0 & \frac{1}{2}dt^2 & 0 \\ 0 & \frac{1}{4}dt^4 & 0 & \frac{1}{2}dt^3 & 0 & \frac{1}{2}dt^2 \\ \frac{1}{2}dt^3 & 0 & dt^2 & 0 & dt & 0 \\ 0 & \frac{1}{2}dt^3 & 0 & dt^2 & 0 & dt \\ \frac{1}{2}dt^2 & 0 & dt & 0 & 1 & 0 \\ 0 & \frac{1}{2}dt^2 & 0 & dt & 0 & 1 \end{pmatrix} \sigma_v^2, \quad R = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \phi \quad (8)$$

where σ_v^2 is set to 1 as a hyperparameter, while ϕ is set to 4, corresponding to the squared standard deviation of the lip-landmark detector outputs. These values provided stable and accurate tracking in our experiments.

The Kalman filter tracks the state estimate $\hat{s}_i^t \in \mathbb{R}^6$ and its error covariance P_i^t .

In the predict step:

$$\hat{s}_i^t = A \hat{s}_i^{t-1} \quad (9)$$

$$\hat{P}_i^t = A P_i^{t-1} A^T + Q \quad (10)$$

In the update step, the Kalman gain K_i^t refines the prediction:

$$K_i^t = \hat{P}_i^t H^T (H \hat{P}_i^t H^T + R)^{-1} \quad (11)$$

$$\hat{s}_i^t = \bar{s}_i^t + K_i^t (p_i^t - H \bar{s}_i^t) \quad (12)$$

$$\hat{p}_i^t = (I - K_i^t H) \hat{p}_i^t \quad (13)$$

The final predicted landmark is:

$$\hat{p}_i^t = H \hat{s}_i^t \quad (14)$$

3.6. Talking Face Video Synthesis

Given a set of target landmarks $\{t_i^t\}_{i=1}^{20}$, a latent code w , and an input face image I_o , we synthesize each frame through an iterative optimization process. For the current frame, the FaceAlignment [63] landmark detector is first applied to the input image to obtain an initial set of handles $\{p_i^t\}$, which serve as the starting point for the generation pipeline. The motion supervision stage (Section 3.3) then updates the latent code from w to w' , according to the displacement between the current handle landmarks and the target landmarks, by optimizing the intermediate feature map F . This produces a new feature map F' and a new image I' . Since the handle landmark positions are not provided before the next optimization step, FaceAlignment is reapplied to I' . To mitigate errors and noise introduced by the landmark detector, the detected landmarks are tracked and refined using a constant-acceleration model Kalman filter, yielding corrected landmark positions $\{\hat{p}_i^t\}$. The refined landmarks are then used as updated handles for the next motion supervision step. This process iterates up to 10 optimization steps or stops once the landmarks align with the targets. The same Kalman filter is used both within a frame (for iterative refinement) and across consecutive frames, enabling temporally smooth and stable landmark motion throughout the generated video. An overview of the complete generation pipeline is illustrated in Figure 1.

4. Experiments

4.1. Experimental Setup

We implement our method in PyTorch [71], optimizing the latent code w using Adam [72] with a learning rate of 2×10^{-3} . We set $\lambda = 200$ and radius $r = 3$. For applicable performance, optimization is capped at 10 iterations per frame (typically converging in 5). For pre-processing, GAN inversion is performed using PTI [27] to obtain the latent code w and generator G for the input image, which are used in the generation pipeline (Section 3.1). We then detect landmarks in the first frame using FaceAlignment [63] to initialize the handle landmarks (Equation (1)). Target landmarks are first aligned to the coordinate system of the first frame using the Umeyama algorithm [64] (Section 3.1).

We employ two Kalman filters with the same constant-acceleration formulation described in Section 3.5, applied at different stages of the pipeline. Since target landmarks are obtained from external sources, they may contain frame-to-frame noise that must be smoothed to enable stable animation. The first Kalman filter smooths the target landmark sequence prior to synthesis, with a time step of $dt = \frac{1}{30}$, corresponding to the input frame rate. The second Kalman filter tracks handle landmarks detected on synthesized images during frame generation (the point tracking stage, Section 3.4) using a smaller time step $dt = \frac{1}{30} \times \frac{1}{7}$, reflecting the nominal intra-frame time step associated with the inner optimization iterations performed within each frame. Our method generates 1024×1024 video frames at approximately 0.5s per frame on a single NVIDIA GeForce RTX 2080 Ti GPU (CUDA 11.4) using an unoptimized PyTorch implementation. This hardware setup

represents a common yet limited-resource workstation baseline. Of this time, roughly 25% is spent on motion supervision and the remaining 75% on point tracking.

To evaluate lip movement accuracy and video quality, we report the following metrics:

- Fréchet Video Distance (FVD) [73] measures video realism by comparing the distributions of generated and real video features extracted using a pre-trained I3D network [74] applied to sequences of 20 frames. Let μ_g, Σ_g and μ_r, Σ_r denote the mean and covariance of the generated and real video feature distributions, respectively. FVD is computed as:

$$\text{FVD} = \|\mu_g - \mu_r\|_2^2 + \text{Tr}(\Sigma_g + \Sigma_r - 2(\Sigma_g \Sigma_r)^{1/2}) \quad (15)$$

Lower FVD values indicate higher video realism.

- Average Expression Distance (AED) and Average Pose Distance (APD) [11] assess motion transfer accuracy by comparing 3D Morphable Model (3DMM) parameters between generated and reference frames. For each frame t and feature vector x_t , they are computed as:

$$\text{AED/APD} = \frac{1}{T} \sum_{t=1}^T \|x_t^{\text{gen}} - x_t^{\text{ref}}\|_2 \quad (16)$$

Lower AED and APD values indicate better preservation of expression and pose.

- Cosine similarity (CSIM) evaluates identity preservation by comparing ArcFace [75] embeddings of generated (f_{gen}) and reference (f_{ref}) images:

$$\text{CSIM} = \frac{f_{\text{gen}} \cdot f_{\text{ref}}}{\|f_{\text{gen}}\|_2 \|f_{\text{ref}}\|_2} \quad (17)$$

Higher CSIM values indicate better identity preservation.

4.2. Lip Landmark-Driven Talking Face Generation

We compare our method with MakeItTalk [7], DragGAN [28], and FreeDrag [29]—all driven by lip landmarks. MakeItTalk uses an LSTM [76] to convert audio into facial landmarks, which are then used to animate a static image via warping or image translation. DragGAN relies on nearest-neighbor similarity for point tracking in latent space, while FreeDrag improves tracking via adaptive template feature updates and line search with backtracking. For both, we adopt their tracking methods within our framework for comparison.

For each method, we generate a video from a single image and a lip-landmark stream, evaluating them on the HDTF dataset [13], which contains 20 videos of different individuals. We assess two tasks: same-identity reenactment, where source and driving landmarks come from the same identity, and cross-identity generation, where they differ, presenting greater challenges due to facial structure variation. For same-identity, the first frame of each video serves as input; lip landmarks are extracted from the first 400 frames of 20 HDTF test videos [63], producing 8,000 frames with ground truth. For cross-identity, we animate 16 real portraits from the Face Research Lab London Set [77] using the 20 HDTF videos, yielding 128,000 synthetic frames. Lacking ground truth, we assess identity preservation via cosine similarity (CSIM), and evaluate expression and pose accuracy using AED and APD between driving and synthetic images. Video quality is measured with FVD against the 16 identity-swapped versions of each HDTF video.

4.3. Qualitative Evaluation

Figure 2 shows same-identity results. Each row shows a different identity, and columns correspond to the source image, driving image, MakeItTalk, FreeDrag, DragGAN, and our method, GANimate, respectively. While all methods modify the source image, clear

differences in lip motion fidelity and visual quality are observed. FreeDrag often exhibits limited mouth articulation, with outputs remaining visually close to the source image and failing to reproduce the mouth opening and shape changes present in the driving frames. As a result, some FreeDrag outputs appear nearly identical to the source. MakeItTalk produces visible lip motion but lacks detail in mouth shapes and teeth and does not preserve identity and facial structure. DragGAN generates stronger mouth deformations but frequently introduces distorted lip geometry and inconsistent mouth shapes. In contrast, our method consistently produces clear and expressive lip motion that closely follows the driving expressions, maintains coherent lip shapes, renders realistic teeth (e.g., first and last rows), and preserves facial identity across diverse source–driving pairs.



Figure 2. Qualitative comparisons on the task of same-identity animation.

Figure 3 presents cross-identity animation results. Our method demonstrates superior expression accuracy and visual realism, successfully modeling open-mouth motions from closed-mouth source images while preserving lip structure and generating natural-looking teeth (top two rows). FreeDrag, again, exhibits limited lip articulation and fails to open the mouth or change the expression according to the driving frames, resulting in outputs that remain close to the source image. MakeItTalk produces visible motion but lacks fine-grained detail and struggles to preserve facial identity, whereas DragGAN tends to exaggerate mouth deformations. As illustrated in the last row, our method maintains coherent lip expressions and structure, while competing approaches frequently collapse the mouth or fail to accurately track the driving motion. Overall, our method consistently delivers the most realistic, coherent results, preserving both expression dynamics and identity—further demonstrated in the supplementary demo.

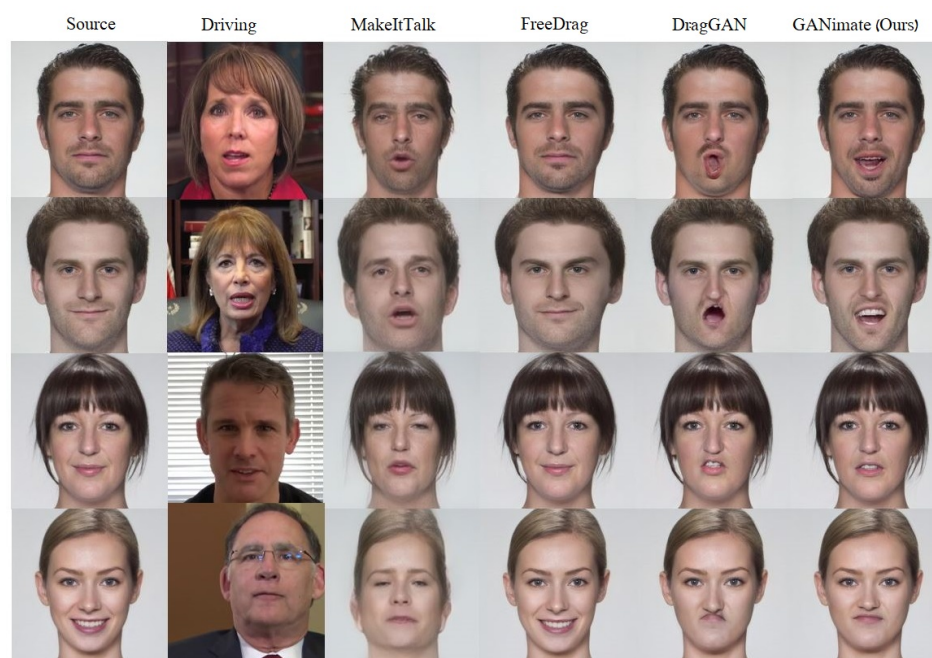


Figure 3. Qualitative comparisons on the task of cross-identity animation.

4.4. Quantitative Evaluation

Tables 1 and 2 show results for same-ID and cross-ID experiments. In the same-ID case, our method achieves the best FVD, AED, and APD scores, reflecting superior quality and realism and accurate expression and pose preservation. It also maintains a high CSIM, indicating strong identity preservation. In the cross-ID case, our method again yields the best FVD, with effective expression and pose tracking. Although it trades some precision for smoother lip motion due to our filtering mechanism, it still offers competitive identity preservation. While MakeItTalk reports the highest CSIM, our approach balances expression accuracy and temporal coherence more effectively. Notably, GANimate achieves the lowest FVD in both settings, highlighting its superior temporal coherence and overall realism, which is the most critical metric for evaluating talking face animation quality.

High parametric efficiency from 2D lip landmarks: By combining a lightweight generative architecture with a Kalman filter-based detection and tracking mechanism, the method achieves superior temporal coherence and stability compared to existing tracking methods.

Table 1. Comparison on HDTF dataset (same-ID case). GANimate achieves the best FVD (temporal coherence and realism), AED, and APD (expression and pose synchronization). While FreeDrag [29] attains the highest CSIM (identity preservation), GANimate achieves the best balance with realistic, high-quality animations.

Method	FVD ₂₀ ↓	AED ↓	APD ↓	CSIM ↑
MakeItTalk	718.04	1.81	0.164	0.741
DragGAN	461.96	2.87	0.129	0.596
FreeDrag	694.41	3.23	0.120	0.822
GANimate (Ours)	375.02	1.68	0.115	0.706

Table 2. Comparison on the HDTF dataset (cross-ID). Results highlight the trade-off between temporal coherence and realism (FVD) versus expression and pose tracking (AED, APD). GANimate achieves the best FVD, balancing these factors well. MakeItTalk excels in AED and CSIM, while DragGAN has the lowest APD.

Method	FVD ₂₀ ↓	AED ↓	APD ↓	CSIM ↑
MakeItTalk	762.35	2.36	0.128	0.674
DragGAN	649.83	2.73	0.121	0.502
FreeDrag	865.45	2.89	0.134	0.615
GANimate (Ours)	429.59	2.65	0.126	0.571

5. Discussion

GANimate demonstrates that talking face animation can be achieved with high efficiency from 2D lip landmarks. Our method combines a lightweight model of 56M parameters for 1024 resolution and 30M parameters for smaller resolutions with generative models and a novel Kalman filter-based detection and tracking mechanism. This approach outperforms existing tracking methods in temporal coherence and stability, producing realistic animations within a simple yet effective architecture. Our results demonstrate strong performance and provide convincing results for both same-ID and cross-ID animations.

While effective in both settings, GANimate relies on only 20 lip landmarks. This limits its ability to capture the full range of facial deformations compared to denser 68-point or 3D models such as [78]. Additionally, the model primarily animates the lower face, and future work will aim to extend this to the entire face to enhance expressiveness and naturalness.

Furthermore, although the current implementation achieves high-fidelity 1024 × 1024 generation, the 0.5 s latency on a standard desktop GPU indicates that its current form is yet to be optimized for real-time applications. This baseline represents a common, limited-resource workstation. To enable mobile avatar animation, computational overhead could be reduced by optimizing the implementation for target hardware and using smaller image resolutions, such as 512 × 512 or 256 × 256. Given the lightweight architecture (30–56M parameters), these optimizations are expected to allow real-time performance on mobile GPUs, enabling interactive avatars for video communication and digital assistants.

6. Conclusions

We present GANimate, a novel method for talking face generation from 2D lip landmarks, combining GAN-based latent space optimization with a classical Kalman filter for detection and tracking. Our new tracking mechanism outperforms existing methods in temporal coherence and stability, ensuring smoother and more realistic animations. GANimate leverages both learning- and model-based techniques in a compact and efficient architecture, seamlessly integrating with any lip-landmark generator and adaptively refining animation quality during inference. Experimental results show that GANimate achieves the best temporal coherence and realism, producing natural and lifelike talking face animations. Its efficiency and modularity make it particularly suitable for mobile edge devices, paving the way for practical deployment in interactive applications.

Supplementary Materials: The following supporting information can be downloaded at <https://www.mdpi.com/article/10.3390/s26041377/s1>. Video S1: A demonstration of GANimate illustrating the generated talking face animation.

Author Contributions: Conceptualization, D.R. and E.F.; methodology, E.F. and B.O.; software, E.F.; validation E.F.; formal analysis, E.F.; investigation, E.F.; resources, D.R.; data curation, E.F.; writing—original draft preparation, E.F.; writing—review and editing, E.F. and D.R.; visualization, E.F.; supervision, D.R.; project administration, D.R.; funding acquisition, D.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Xu, S.; Chen, G.; Guo, Y.X.; Yang, J.; Li, C.; Zang, Z.; Zhang, Y.; Tong, X.; Guo, B. Vasa-1: Lifelike audio-driven talking faces generated in real time. *arXiv* **2024**, arXiv:2404.10667.
2. Wang, D.; Deng, Y.; Yin, Z.; Shum, H.Y.; Wang, B. Progressive disentangled representation learning for fine-grained controllable talking head synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 17979–17989.
3. Girdhar, R.; Singh, M.; Brown, A.; Duval, Q.; Azadi, S.; Rambhatla, S.S.; Shah, A.; Yin, X.; Parikh, D.; Misra, I. Emu video: Factorizing text-to-video generation by explicit image conditioning. *arXiv* **2023**, arXiv:2311.10709.
4. Stypułkowski, M.; Vougioukas, K.; He, S.; Zięba, M.; Petridis, S.; Pantic, M. Diffused heads: Diffusion models beat gans on talking-face generation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2024; pp. 5091–5100.
5. Zhang, B.; Qi, C.; Zhang, P.; Zhang, B.; Wu, H.; Chen, D.; Chen, Q.; Wang, Y.; Wen, F. Metaportrait: Identity-preserving talking head generation with fast personalized adaptation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 22096–22105.
6. Pang, Y.; Zhang, Y.; Quan, W.; Fan, Y.; Cun, X.; Shan, Y.; Yan, D.m. Dpe: Disentanglement of pose and expression for general video portrait editing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 427–436.
7. Zhou, Y.; Han, X.; Shechtman, E.; Echevarria, J.; Kalogerakis, E.; Li, D. Makeltalk: Speaker-aware talking-head animation. *ACM Trans. Graph. (TOG)* **2020**, *39*, 221.
8. Zakharov, E.; Shysheya, A.; Burkov, E.; Lempitsky, V. Few-shot adversarial learning of realistic neural talking head models. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9459–9468.
9. Siarohin, A.; Lathuilière, S.; Tulyakov, S.; Ricci, E.; Sebe, N. First order motion model for image animation. In *Proceedings of the Advances in Neural Information Processing Systems*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019, Volume 32.
10. Doukas, M.C.; Zafeiriou, S.; Sharmanska, V. Headgan: One-shot neural head synthesis and editing. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 14398–14407.
11. Ren, Y.; Li, G.; Chen, Y.; Li, T.H.; Liu, S. Pirenderer: Controllable portrait image generation via semantic neural rendering. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 13759–13768.
12. Wu, H.; Jia, J.; Wang, H.; Dou, Y.; Duan, C.; Deng, Q. Imitating arbitrary talking style for realistic audio-driven talking face synthesis. In Proceedings of the 29th ACM International Conference on Multimedia, Chengdu, China, 20–24 October 2021; pp. 1478–1486.
13. Zhang, Z.; Li, L.; Ding, Y.; Fan, C. Flow-guided one-shot talking face generation with a high-resolution audio-visual dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 3661–3670.
14. Yin, F.; Zhang, Y.; Cun, X.; Cao, M.; Fan, Y.; Wang, X.; Bai, Q.; Wu, B.; Wang, J.; Yang, Y. Styleheat: One-shot high-resolution editable talking face generation via pre-trained stylegan. In *Proceedings of the European Conference on Computer Vision*; Springer: Cham, Switzerland, 2022; pp. 85–101.

15. Yi, R.; Ye, Z.; Zhang, J.; Bao, H.; Liu, Y.J. Audio-driven talking face video generation with learning-based personalized head pose. *arXiv* **2020**, arXiv:2002.10137.
16. Zhang, C.; Zhao, Y.; Huang, Y.; Zeng, M.; Ni, S.; Budagavi, M.; Guo, X. Facial: Synthesizing dynamic talking face with implicit attribute learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 3867–3876.
17. Du, C.; Chen, Q.; He, T.; Tan, X.; Chen, X.; Yu, K.; Zhao, S.; Bian, J. Dae-talker: High fidelity speech-driven talking face generation with diffusion autoencoder. In Proceedings of the 31st ACM International Conference on Multimedia, Ottawa, ON, Canada, 29 October–3 November 2023; pp. 4281–4289.
18. Tian, L.; Wang, Q.; Zhang, B.; Bo, L. EMO: Emote Portrait Alive Generating Expressive Portrait Videos with Audio2Video Diffusion Model Under Weak Conditions. In *Proceedings of the European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2025; pp. 244–260.
19. Tian, L.; Hu, S.; Wang, Q.; Zhang, B.; Bo, L. EMO2: End-Effector Guided Audio-Driven Avatar Video Generation. *arXiv* **2025**, arXiv:2501.10687.
20. Blattmann, A.; Dockhorn, T.; Kulal, S.; Mendelevitch, D.; Kilian, M.; Lorenz, D.; Levi, Y.; English, Z.; Voleti, V.; Letts, A.; et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv* **2023**, arXiv:2311.15127.
21. Brooks, T.; Peebles, B.; Holmes, C.; DePue, W.; Guo, Y.; Jing, L.; Schnurr, D.; Taylor, J.; Luhman, T.; Luhman, E.; et al. Video Generation Models as World Simulators. 2024. Available online: <https://openai.com/research/video-generation-models-as-world-simulators> (accessed on 3 January 2026)
22. Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; Aila, T. Analyzing and improving the image quality of stylegan. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 8110–8119.
23. Abdal, R.; Zhu, P.; Mitra, N.J.; Wonka, P. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Trans. Graph. (ToG)* **2021**, *40*, 21.
24. Tewari, A.; Elgharib, M.; Bharaj, G.; Bernard, F.; Seidel, H.P.; Pérez, P.; Zollhofer, M.; Theobalt, C. Stylerig: Rigging stylegan for 3d control over portrait images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6142–6151.
25. Wang, T.; Zhang, Y.; Fan, Y.; Wang, J.; Chen, Q. High-fidelity gan inversion for image attribute editing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11379–11388.
26. Zhu, P.; Abdal, R.; Femiani, J.; Wonka, P. Barbershop: Gan-based image compositing using segmentation masks. *arXiv* **2021**, arXiv:2106.01505.
27. Roich, D.; Mokady, R.; Bermano, A.H.; Cohen-Or, D. Pivotal tuning for latent-based editing of real images. *ACM Trans. Graph. (TOG)* **2022**, *42*, 6.
28. Pan, X.; Tewari, A.; Leimkühler, T.; Liu, L.; Meka, A.; Theobalt, C. Drag your gan: Interactive point-based manipulation on the generative image manifold. In Proceedings of the ACM SIGGRAPH 2023 Conference Proceedings, Los Angeles, CA, USA, 6–10 August 2023; pp. 1–11.
29. Ling, P.; Chen, L.; Zhang, P.; Chen, H.; Jin, Y. Freedrag: Point tracking is not you need for interactive point-based image editing. *arXiv* **2023**, arXiv:2307.04684.
30. Welch, G.; Bishop, G. *An Introduction to the Kalman Filter*; Technical Report TR 95-041; University of North Carolina at Chapel Hill: Chapel Hill, NC, USA, 1995.
31. Prajwal, K.; Mukhopadhyay, R.; Namboodiri, V.P.; Jawahar, C. A lip sync expert is all you need for speech to lip generation in the wild. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; pp. 484–492.
32. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 10684–10695.
33. Wu, W.; Zhang, Y.; Li, C.; Qian, C.; Loy, C.C. Reenactgan: Learning to reenact faces via boundary transfer. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 603–619.
34. Bansal, A.; Ma, S.; Ramanan, D.; Sheikh, Y. Recycle-gan: Unsupervised video retargeting. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 119–135.
35. Wang, T.C.; Liu, M.Y.; Zhu, J.Y.; Liu, G.; Tao, A.; Kautz, J.; Catanzaro, B. Video-to-video synthesis. *arXiv* **2018**, arXiv:1808.06601.
36. Wang, T.C.; Liu, M.Y.; Tao, A.; Liu, G.; Kautz, J.; Catanzaro, B. Few-shot video-to-video synthesis. *arXiv* **2019**, arXiv:1910.12713.
37. Wang, Y.; Yang, D.; Bremond, F.; Dantcheva, A. Latent image animator: Learning to animate images via latent space navigation. *arXiv* **2022**, arXiv:2203.09043.
38. Burkov, E.; Pasechnik, I.; Grigorev, A.; Lempitsky, V. Neural head reenactment with latent pose descriptors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 13786–13795.

39. Siarohin, A.; Woodford, O.J.; Ren, J.; Chai, M.; Tulyakov, S. Motion representations for articulated animation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13653–13662.
40. Wang, T.C.; Mallya, A.; Liu, M.Y. One-Shot Free-View Neural Talking-Head Synthesis for Video Conferencing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 10039–10049.
41. Siarohin, A.; Lathuilière, S.; Tulyakov, S.; Ricci, E.; Sebe, N. Animating arbitrary objects via deep motion transfer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019; pp. 2377–2386.
42. Booth, J.; Roussos, A.; Zafeiriou, S.; Ponniah, A.; Dunaway, D. A 3d morphable model learnt from 10,000 faces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5543–5552.
43. Kim, H.; Garrido, P.; Tewari, A.; Xu, W.; Thies, J.; Niessner, M.; Pérez, P.; Richardt, C.; Zollhöfer, M.; Theobalt, C. Deep video portraits. *ACM Trans. Graph. (TOG)* **2018**, *37*, 163.
44. Kim, H.; Elgharib, M.; Zollhöfer, M.; Seidel, H.P.; Beeler, T.; Richardt, C.; Theobalt, C. Neural style-preserving visual dubbing. *ACM Trans. Graph. (TOG)* **2019**, *38*, 178.
45. Suwajanakorn, S.; Seitz, S.M.; Kemelmacher-Shlizerman, I. Synthesizing obama: Learning lip sync from audio. *ACM Trans. Graph. (TOG)* **2017**, *36*, 95.
46. Thies, J.; Elgharib, M.; Tewari, A.; Theobalt, C.; Nießner, M. Neural voice puppetry: Audio-driven facial reenactment. In *Computer Vision—ECCV 2020, Proceedings of the 16th European Conference, Glasgow, UK, 23–28 August 2020*; Proceedings, Part XVI 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 716–731.
47. Wen, X.; Wang, M.; Richardt, C.; Chen, Z.Y.; Hu, S.M. Photorealistic audio-driven video portraits. *IEEE Trans. Vis. Comput. Graph.* **2020**, *26*, 3457–3466.
48. Chung, J.S.; Zisserman, A. Out of time: Automated lip sync in the wild. In *Computer Vision—ACCV 2016 Workshops, Proceedings of the ACCV 2016 International Workshops, Taipei, Taiwan, 20–24 November 2016*; Revised Selected Papers, Part II 13; Springer: Cham, Switzerland, 2017; pp. 251–263.
49. Zhou, L.; Chen, Z.; Cao, Y.; Zhang, S.; Zeng, M.; Zhang, Q. Talking Face Generation by Adversarially Disentangled Audio-Visual Representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 5070–5079.
50. Zhou, H.; Sun, Y.; Wu, W.; Loy, C.C.; Wang, X.; Liu, Z. Pose-controllable talking face generation by implicitly modularized audio-visual representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2021; pp. 4176–4186.
51. Song, L.; Wu, W.; Fu, C.; Qian, C.; Loy, C.C.; He, R. Everything’s Talkin’: Pareidolia Face Reenactment. *arXiv* **2021**, arXiv:2104.03061.
52. Guo, J.; Zhang, D.; Liu, X.; Zhong, Z.; Zhang, Y.; Wan, P.; Zhang, D. Liveportrait: Efficient portrait animation with stitching and retargeting control. *arXiv* **2024**, arXiv:2407.03168.
53. Lahiri, A.; Kwatra, V.; Frueh, C.; Lewis, J.; Bregler, C. Lipsync3d: Data-efficient learning of personalized 3d talking faces from video using pose and lighting normalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2755–2764.
54. Leimkühler, T.; Drettakis, G. Freestylegan: Free-view editable portrait rendering with the camera manifold. *arXiv* **2021**, arXiv:2109.09378.
55. Patashnik, O.; Wu, Z.; Shechtman, E.; Cohen-Or, D.; Lischinski, D. Styleclip: Text-driven manipulation of stylegan imagery. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2085–2094.
56. Shen, Y.; Gu, J.; Tang, X.; Zhou, B. Interpreting the latent space of gans for semantic face editing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9243–9252.
57. Härkönen, E.; Hertzmann, A.; Lehtinen, J.; Paris, S. Ganspace: Discovering interpretable gan controls. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9841–9850.
58. Shen, Y.; Zhou, B. Closed-form factorization of latent semantics in gans. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 1532–1540.
59. Zhu, J.; Yang, C.; Shen, Y.; Shi, Z.; Dai, B.; Zhao, D.; Chen, Q. Linkgan: Linking gan latents to pixels for controllable image synthesis. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 7656–7666.
60. Wang, S.Y.; Bau, D.; Zhu, J.Y. Rewriting geometric rules of a GAN. *ACM Trans. Graph.* **2022**, *41*, 73.
61. Endo, Y.; Yamaguchi, K. User-Controllable Latent Transformer for Talking Head Synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 18042–18051.
62. Jeong, D.; Lee, I.; Paik, J. Audio-to-Facial Landmarks Generator for Talking Face Video Synthesis. In Proceedings of the International Conference on Electronics, Information, and Communication (ICEIC), Singapore, 5–8 February 2023; pp. 1–2.

63. Bulat, A.; Tzimiropoulos, G. How far are we from solving the 2D & 3D Face Alignment problem? (and a dataset of 230,000 3D facial landmarks). In Proceedings of the International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
64. Umeyama, S. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 376–380.
65. Brox, T.; Malik, J. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 500–513.
66. Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; Van Der Smagt, P.; Cremers, D.; Brox, T. FlowNet: Learning optical flow with convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2758–2766.
67. Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; Brox, T. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2462–2470.
68. Teed, Z.; Deng, J. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision—ECCV 2020, Proceedings of the 16th European Conference, Glasgow, UK, 23–28 August 2020*; Proceedings, Part II 16; Springer: Cham, Switzerland, 2020; pp. 402–419.
69. Harley, A.W.; Fang, Z.; Fragkiadaki, K. Particle video revisited: Tracking through occlusions using point trajectories. In *Proceedings of the European Conference on Computer Vision*; Springer: Cham, Switzerland, 2022; pp. 59–75.
70. Prabhu, U.; Seshadri, K.; Savvides, M. Automatic facial landmark tracking in video sequences using kalman filter assisted active shape models. In *Trends and Topics in Computer Vision, Proceedings of the ECCV 2010 Workshops, Heraklion, Crete, Greece, 10–11 September 2010*; Revised Selected Papers, Part I 11; Springer: Berlin/Heidelberg, Germany, 2012; pp. 86–99.
71. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Raison, M.; Antiga, A.; Lerer, A. Automatic Differentiation in PyTorch. In Proceedings of the Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017.
72. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
73. Unterthiner, T.; van Steenkiste, S.; Kurach, K.; Esser, P.; Heigold, G.; Kalchbrenner, N.; Bousquet, O.; Scholkopf, B.; Riedmiller, M. Towards Accurate Generative Models of Video: A New Metric and Challenges. *arXiv* **2019**, arXiv:1812.01717.
74. Carreira, J.; Zisserman, A. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2017; pp. 6299–6308.
75. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4690–4699.
76. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780.
77. Lab, F.R. Face Research Lab London Set. 2020. Available online: https://figshare.com/articles/dataset/Face_Research_Lab_London_Set/5047666 (accessed on 3 November 2025)
78. Kartynnik, Y.; Ablavatski, A.; Grishchenko, I.; Grundmann, M. Real-time facial surface geometry from monocular video on mobile GPUs. *arXiv* **2019**, arXiv:1907.06724.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.