University of Salerno

Department of Information and Electrical Engineering
and Applied Mathematics

Master's Degree in Computer Engineering

Curriculum in Artificial Intelligence and Intelligent Robotics

Analyzing the Impact of Semantic Temporal Conditioning in Emotion-Driven Facial Video Generation Using Diffusion Models and GANs

Supervisors

Prof. Mario Vento

Prof. Antonio Greco

Prof. Nicola Strisciuglio

Candidate

Jacopo Volpe

0622702301

Academic Year: 2024/2025

Abstract

Automatic generation of short emotional video sequences from static facial images is a challenging task, as it must jointly ensure temporal coherence, identity preservation, and realistic expression transitions. The problem becomes particularly hard when the goal is to model plausible dynamics over time while keeping the subject identity stable and avoiding visually inconsistent artifacts across frames.

In the context of conditional facial video generation, this thesis builds on the VICEGAN baseline and uses LFDM as a state-of-the-art reference for comparison. In VICEGAN, a recurrent motion-code generator first produces a latent trajectory: a series of motion codes encoding temporal dynamics, which a GAN-based frame generator then converts into video frames conditioned on the target emotion and a neutral input image. The thesis focuses on the semantic role of temporal conditioning codes for improving temporal coherence and controllability of expression dynamics in generated videos, and systematically compares GAN-based and diffusion-based frame generators in terms of frame quality, visual realism and identity preservation.

The first contribution is to quantify the effect of enforcing an emotion-aware semantics in the temporal conditioning: instead of learning generic dynamics from noise, the VICEGAN motion-code generator is explicitly conditioned on the target emotion so that motion codes themselves encode the evolution of the expression. This shifts the main conditioning signal toward the part of the pipeline responsible for dynamics, improving controllability over expression changes and reducing the need for the frame synthesis stage to implicitly infer motion patterns.

The second contribution is a diffusion-based frame synthesis module that replaces the GAN generator in VICEGAN. By combining identity conditioning from a neutral input frame with motion codes and emotion labels inside a conditional

U-Net, the diffusion model aims to improve visual realism, training stability, and identity preservation over time.

Finally, an external Expression Embedding Network is employed to provide compact, semantically meaningful expression representations insensitive to identity. These embeddings support a modular design in which identity, expression, and motion are disentangled: they make it possible to drive the diffusion model with retrieval-based motion trajectories, obtained by reusing motion codes from real training sequences with the target emotion, while preserving identity and temporal coherence in the generated videos.

The system is trained and evaluated on the MUG dataset. Performance is assessed using three groups of measures: a video-quality score based on the Fréchet Video Distance (FVD), emotion-recognition accuracy through Dominant Emotion Accuracy and Average Emotion Accuracy, and identity and temporal consistency through the Average Content Distance for Identity and for temporal Consistency (ACD-I and ACD-C). The results show that moving emotion conditioning to the motion generator improves classifier-based emotion scores over the VICEGAN baseline, while diffusion-based frame synthesis yields substantially better realism according to FVD and stronger identity preservation, outperforming both the baseline and the LFDM reference. The experimental analysis highlights a trade-off between distribution-level realism and classifier-based emotion scores, and is therefore complemented by a user study that directly measures human perception of realism, identity consistency, and emotion plausibility.

Contents

Chapter 1

Introduction

The automatic generation of realistic emotional videos from static facial images represents a challenging problem at the intersection of computer vision, generative modeling, and affective computing. The ability to synthesize natural facial expressions and their temporal evolution has significant applications in human-computer interaction, digital content creation, video conferencing, and entertainment. This thesis addresses the problem of generating temporally coherent video sequences that depict smooth transitions from neutral expressions to specific target emotions while preserving the subject's identity.

Recent advances in generative models, particularly Generative Adversarial Networks (GANs) [12] and Denoising Diffusion Probabilistic Models (DDPMs) [15], have enabled the synthesis of high-quality images and videos. However, generating emotionally expressive facial videos remains challenging due to the need to model complex spatiotemporal dynamics while maintaining consistency across multiple dimensions: visual quality, temporal coherence, identity preservation, and emotional realism.

This thesis addresses the problem of **automatic emotional video generation from static facial images**. Given a neutral facial image and a target emotion label, the objective is to produce a short video sequence depicting a smooth and natural transition from the neutral expression to the specified target emotion.

The main challenges involved in this task include:

- **Spatiotemporal Consistency:** Maintaining both spatial consistency within individual frames and temporal consistency across consecutive frames to ensure smooth video playback without artifacts or discontinu-

ities.

- **Identity Fidelity:** Preserving the subject's identity throughout the entire sequence, ensuring that facial features, structure, and appearance remain recognizable despite changes in expression.
- **Emotional Realism:** Generating facial movements and expressions that appear natural and realistic, following the typical dynamics of human emotional displays rather than abrupt or artificial transitions.
- **Emotion-Motion Coupling:** Ensuring that temporal dynamics (motion patterns) are appropriately conditioned on and reflective of the target emotion, capturing emotion-specific characteristics in facial movements.

These challenges are particularly difficult because facial expressions involve subtle, coordinated movements of multiple facial regions, and different emotions exhibit distinct temporal patterns of onset, apex, and offset phases.

1.1 Research Questions

This thesis seeks to answer the following research questions:

1. How can the model's conditioning strategy be designed so that motion codes acquire emotion-related semantics and yield emotion-specific temporal dynamics?
2. Can diffusion models provide superior frame synthesis quality compared to GAN-based approaches for emotional video generation?
3. Can a pretrained model, whose embeddings already capture emotional semantics, be used to drive facial-expression video generation, transferring the expression dynamics learned from the training set to previously unseen identities?
4. How does the semantic content of the motion codes affect the quality of the generated videos across different model configurations?

1.2 Proposed Approach

This work starts from VICEGAN [4], a two-stage model for emotional video generation. In VICEGAN, a recurrent motion generator first produces a sequence of motion codes, and a frame generator then converts these codes into video frames conditioned on the input face and the target emotion. Starting from this baseline, the thesis studies a simple design question: where should emotion information be injected, and which frame generator should be used. To answer this question, the work explores three main ideas that can also be combined in different settings.

1.2.1 Emotion-Aware Motion Code Generation

The first idea is to move emotion conditioning from the frame generator to the motion generator. Instead of producing generic motion codes, the motion generator is trained to produce motion trajectories that already contain emotion information. In this way, the temporal signal is not only about movement, but also about how the target expression develops over time.

This change is expected to improve control over the dynamics of the generated sequence, because the target emotion affects the motion representation directly.

1.2.2 Diffusion-Based Frame Synthesis

The second idea is to replace the GAN-based frame generator with a diffusion model. Denoising Diffusion Probabilistic Models (DDPMs) are attractive for this task because they often provide:

- Greater training stability compared to adversarial training
- Higher visual quality and fidelity in generated images
- More effective conditioning mechanisms through feature modulation
- Reduced mode collapse and training instabilities

In our framework, the diffusion model uses a conditional U-Net [24]. It receives identity information from the neutral face, motion information from the temporal codes, and emotion information from the target label. The model then generates each frame through an iterative denoising process.

1.2.3 Expression Embedding Network

The third idea is to use a pre-trained Expression Embedding Network based on the compact embedding approach of Vemulapalli and Agarwala [27]. This network produces compact expression embeddings in which distances reflect facial-expression similarity, while being less sensitive to subject identity. The main advantages of this component are:

- A semantically meaningful space for facial-expression similarity
- Compact embeddings that support analysis, retrieval, and conditioning
- A modular workflow based on a frozen, reusable embedding network

In this thesis, the embedding network is kept frozen and used to extract expression embeddings from the training data, which are then used to train the diffusion model supporting retrieval-based conditioning.

1.3 Thesis Contributions

The main contributions of this thesis are:

1. **A clear comparison of conditioning strategies:** this work compares the standard VICEGAN design with a variant in which emotion is moved from the frame generator to the motion generator, showing how this choice changes the generated dynamics.
2. **A diffusion-based frame generator for facial videos:** a conditional diffusion model is adapted to facial expression video generation and compared with the original GAN-based synthesis pipeline.
3. **A modular setting based on expression embeddings:** the study explores how a compact expression embedding space can be used to separate identity, expression, and motion, and to support retrieval-based motion conditioning.
4. **A broad experimental evaluation:** all settings are compared using established metrics, including Fréchet Video Distance (FVD), Dominant Emotion Accuracy (DEA), Average Emotion Accuracy (AEA), Average

Content Distance for Identity (ACD-I), and Average Content Distance for temporal Consistency (ACD-C).

5. **Human validation through a user study:** automatic metrics are complemented with a user study to verify whether the model selected from the quantitative results is also preferred by human observers.

1.4 Thesis Organization

The remainder of this thesis is organized as follows:

- **Chapter 2:** Provides background on video generation, reviewing relevant literature on GANs, diffusion models, spatiotemporal consistency, and disentangled representation learning. It also presents a detailed analysis of the VICEGAN architecture and related generative approaches.
- **Chapter 3:** Presents the proposed methodology in detail, describing the Expression Embedding Network, the Emotion-aware Motion Code Generator, and the Diffusion-based Frame Synthesizer. It explains the integration of these components, training procedures, and specific architectural design choices.
- **Chapter 4:** Describes the experimental setup, including the MUG Facial Expression Database, evaluation metrics, and training configuration.
- **Chapter 5:** Presents the quantitative and qualitative results, including comparisons with baseline methods, ablation studies, and the user study.
- **Chapter 6:** Concludes the thesis with a summary of the main findings, a discussion of contributions and limitations, and suggestions for future research directions.

Chapter 2

Literature Review

Video generation has rapidly evolved as a key component of Artificial Intelligence Generated Content (AIGC), enabling the creation of dynamic visual sequences that extend beyond the capabilities of static image synthesis. While image generation focuses on pixel-level quality within a single frame, video generation introduces the complexity of the time dimension, requiring models to maintain coherence across a sequence of frames.

2.1 Spatiotemporal Consistency in Video Generation

A fundamental challenge in video synthesis is maintaining spatiotemporal consistency. As highlighted in the survey [29], video generation requires satisfying two distinct yet interconnected forms of consistency:

- **Spatial Consistency**: This refers to the quality and structural integrity of individual frames. It ensures that objects, colors, and textures remain coherent and free from varying distortions or artifacts. In the context of facial generation, spatial consistency is critical for preserving the subject's identity and anatomical correctness in every frame.
- **Temporal Consistency**: This pertains to the smoothness of transitions between consecutive frames. It ensures that motion, lighting changes, and object deformations occur naturally over time without abrupt jumps or flickering.

Ensuring high spatiotemporal consistency involves several stages of the generative process, from selecting the foundation models to designing data representations and post-processing techniques. While foundation models provide the mathematical framework for motion, specialized representations are needed to capture both appearance and movement effectively. In facial expression synthesis, balancing these two aspects is particularly challenging: the model must generate dynamic facial movements while keeping the person's identity consistent.

2.2 VICEGAN: Baseline Architecture for Video Generation

VICEGAN [4] (Video Identity-Consistent Emotion GAN) is a framework designed for controllable video generation through a two-stage architecture that separates motion generation from frame synthesis. The system is built on the principle of separating temporal dynamics from spatial content, enabling independent control over motion patterns and visual appearance.

2.2.1 Architecture Overview

The VICEGAN architecture consists of two primary components that operate sequentially:

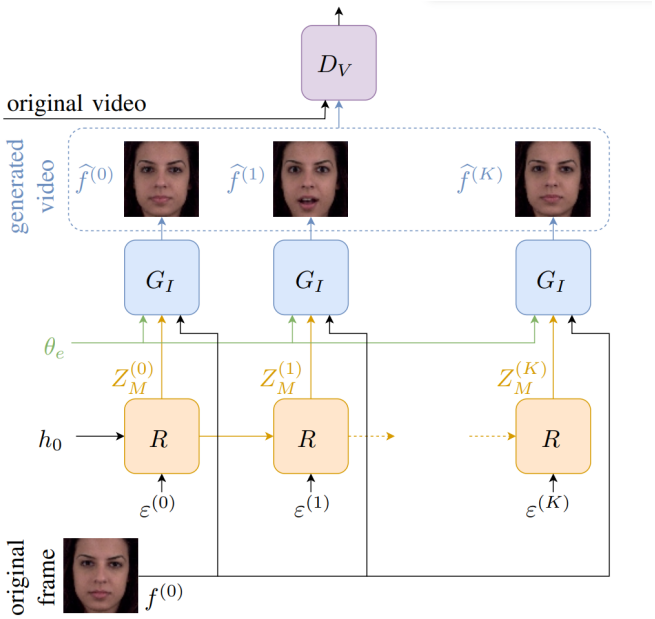


Figure 2.1: VICEGAN architecture overview. Figure adapted from [4].

Motion Code Generator: The first component is a recurrent neural network, implemented using Gated Recurrent Units (GRUs), used for generating a sequence of motion codes. These codes represent temporal dynamics and are produced autoregressively, where each code depends on the previous states in the sequence. The motion code generator takes as input an initial random latent vector and generates a trajectory of latent representations that encode motion information over time.

Frame Generator: The second component is a convolutional neural network that synthesizes individual video frames from the motion codes. This generator operates on each motion code independently, translating the latent motion representation into a corresponding video frame. In VICEGAN, frame synthesis is conditioned on (i) the sequence of motion vectors generated by the RNN, (ii) the target emotion label θ_e, and (iii) the neutral input frame $f^{(0)}$ that provides the identity/content to be preserved along the entire sequence. Importantly, the identity is not provided as an explicit conditioning label to the generator; instead, it is inferred from the input frame and enforced through a dedicated identity-preserving loss.

During generation, VICEGAN produces videos by first sampling or generating a sequence of motion codes through the motion code generator, then feeding each code through the frame generator to produce the corresponding frames. This separation allows for flexible control over the generation process, as motion and appearance can be manipulated independently [4].

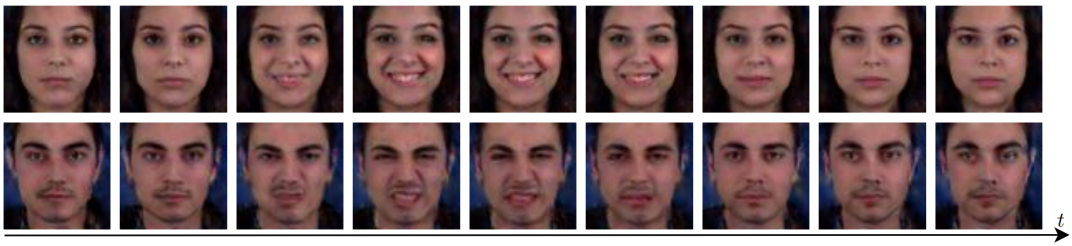


Figure 2.2: Examples of emotional video sequences generated by VICEGAN. Each row shows a sequence of frames expressing a specific target emotion from a neutral input image. Figure from [4].

2.2.2 Training and Loss Functions

The learning problem is cast in the standard **adversarial learning** framework as a min–max game, in which the generator G_I and the RNN R try to fool

the discriminator D_V, while the discriminator tries to correctly distinguish real samples from generated ones:

$$\min_{R,G_I} \max_{D_V} \mathcal{L}(R, G_I, D_V). \tag{2.1}$$

The total loss is composed of three terms:

$$\mathcal{L}(R, G_I, D_V) = \mathcal{L}_{adv} + \lambda_{emo}\,\mathcal{L}_{emo} + \lambda_{ide}\,\mathcal{L}_{ide}, \tag{2.2}$$

where λ_{emo} and λ_{ide} are scalar weights that balance the relative importance of the auxiliary terms.

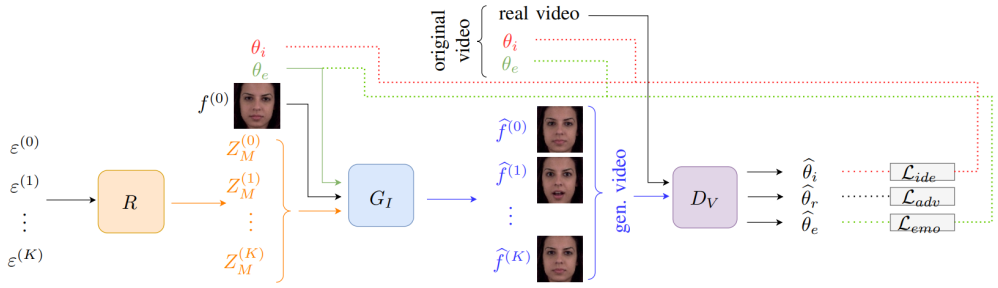


Figure 2.3: VICEGAN training loss components and their interactions. From [4].

- **Adversarial loss** ($\mathcal{L}_{adv}$). The standard GAN loss applied to video sequences, used to enforce *video realism*. It pushes G_I to produce natural-looking frames and drives R to generate coherent motion vectors, free from spatial or temporal anomalies.

- **Emotion classification loss** ($\mathcal{L}_{emo}$). Inspired by InfoGAN [7], this term forces the generator to produce videos belonging to a specific emotion class. It is implemented as a Categorical Cross-Entropy (CCE) between the target emotion θ_e and the emotion predicted by the discriminator $\hat{\theta}_e$. The gradient of this loss flows back through the RNN and updates its weights, so that R learns to map Gaussian noise to motion-vector distributions that consistently represent the target emotion.

- **Identity-preserving loss** ($\mathcal{L}_{ide}$). An original component introduced by the authors to force the model to preserve the identity of the input face and keep it consistent throughout the video. It uses the identity output $\hat{\theta}_i$ of the discriminator and compares it (via CCE) with the expected identity,

which is inferred directly from the input frame $f^{(0)}$. This loss is essential for mitigating the spatio-temporal artifacts typical of two-stage pipelines.

Training relies on the **multi-task discriminator** D_V, whose multiple output heads (real/fake, emotion class, identity class) allow the different requirements (realism, emotion accuracy, and identity preservation) to be balanced within a single network.

The procedure follows these cyclic steps:

1. **Discriminator update.** For each video in the batch, D_V analyses both the real sample and the corresponding generated sample. The total discriminator loss $\mathcal{L}_D$ is computed and its weights are updated via the **AdamW** optimiser. Note that the classification heads (emotion and identity) are trained with CCE *only on real videos*, since the labels for generated sequences are not reliable.

2. **Generator and RNN update.**

   - A random target emotion θ_e is sampled and the real first frame $f^{(0)}$ is taken from the dataset.
   - R and G_I generate a synthetic video.
   - D_V evaluates the generated video and provides feedback on realism, emotion, and identity.
   - The generator loss $\mathcal{L}_G$ is computed and the weights of both G_I and R are updated simultaneously via AdamW. Because the generators have no effect on the real-data terms, $\mathcal{L}_G$ includes only the terms that involve generated samples.

2.2.3 Observations and Opportunities for Enhancement

VICEGAN provides an effective framework for video generation through its two-stage architecture. An analysis of its design reveals different opportunities for further improvements.

Motion Code Generation and Emotional Conditioning

In the VICEGAN architecture, motion codes are generated by the RNN without taking the target emotion label as an explicit input. The emotional information

is instead provided to the frame generator, which conditions the synthesis of each frame on the desired emotion. At the same time, the training objective includes an emotion classification loss whose gradient flows through the full pipeline and updates the RNN parameters. However, since the emotion label θ_e is not an input to the RNN and the motion-code generation is initialized only from random noise, the supervision received by R is indirect. In practice, this suggests that the RNN is primarily encouraged to learn *generic expression dynamics* (i.e., the temporal evolution of facial motion), while it may struggle to reliably learn distinct, emotion-specific temporal patterns for each individual emotion class.

This design choice offers modularity and flexibility in the generation process. An interesting direction for exploration would be to investigate whether integrating emotional information directly into the motion code generation could further improve controllability and help capture emotion-specific temporal patterns, such as gradual onset or characteristic dynamics, more explicitly.

Analysis of Motion Code Behavior

To better understand the role of motion codes in VICEGAN, the temporal evolution of motion latent sequences can be visualized. Figure 2.4 shows multiple examples of motion code sequences generated by the RNN. Each visualization represents the temporal evolution of a single sequence, where columns correspond to time steps (16 steps total), rows represent latent dimensions (10 dimensions), and color intensity indicates normalized activation values ranging from 0 to 1.

Overall, these visualizations suggest that motion codes capture smooth temporal transitions and maintain consistency across frames, encoding primarily temporal information such as frame continuity and motion trajectories.

In Chapter 3, we build upon these observations to motivate and introduce the architectural modifications and improvements proposed in this thesis.

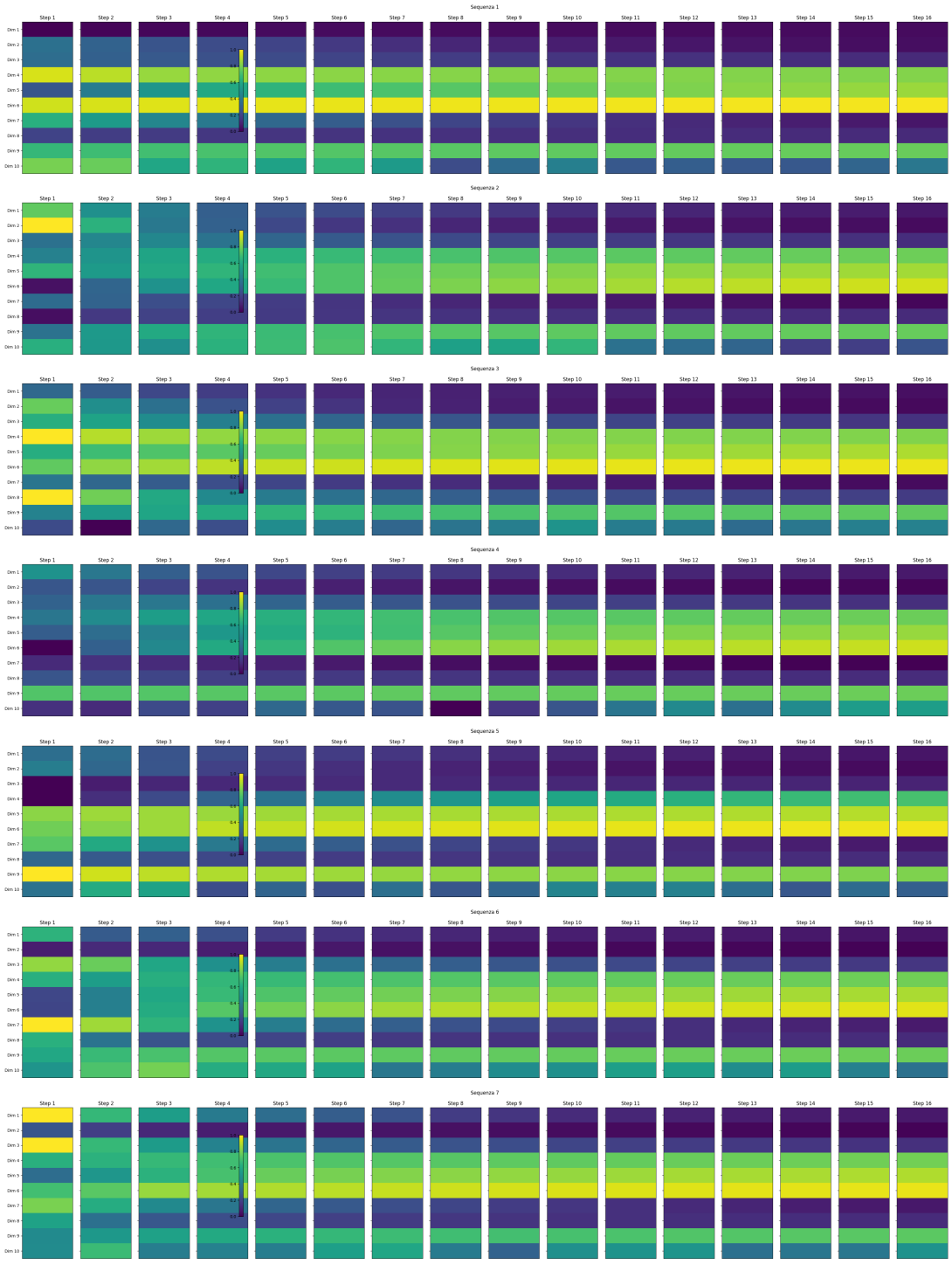


Figure 2.4: Visualization of motion code sequences generated by VICEGAN's RNN. Each row shows a different sequence of 16 time steps across 10 latent dimensions. The sequences were generated using the motion code generator with random initial latent vectors, then normalized using MinMaxScaler across all sequences for consistent visualization. Each heatmap displays how the latent dimensions evolve over time, revealing the temporal patterns encoded in the motion representation.

2.3 Latent Flow Diffusion Model (LFDM)

As a state-of-the-art reference for conditional image-to-video generation from a single input frame, the *Latent Flow Diffusion Model (LFDM)* [20] adopts a latent optical-flow formulation rather than synthesizing pixels directly. The key idea is to generate temporally coherent sequences of *latent optical flows* (together with occlusion maps) that are used to *warp* the initial image across time, which helps preserve fine spatial details while improving temporal coherence. LFDM is trained in two main stages: (i) a Latent Flow Auto-Encoder (LFAE) is learned in an unsupervised manner, where a flow predictor estimates latent motion and an occlusion map between pairs of frames so that the decoder can reconstruct missing or newly revealed regions; (ii) a diffusion model based on a *3D U-Net* is trained to generate sequences of flows and occlusions conditioned on both the initial image (encoded into a latent map) and an external condition (e.g., a class/action label represented via an embedding). At inference time, the diffusion model starts from Gaussian noise to produce a sequence of latent flows, the initial frame is encoded and warped frame-by-frame using the generated motion, and the decoder maps the resulting warped latent representations back to the final video frames. This design brings practical benefits over GAN-based approaches and pixel-space diffusion: it preserves appearance by reusing the spatial content of the input via warping, it is more computationally efficient because diffusion operates in a low-dimensional motion space, it avoids the accumulation of artifacts by deforming the original input x_0 rather than previously synthesized frames, it can be adapted to new domains by fine-tuning the decoder without retraining the whole framework, and it improves temporal consistency thanks to the 3D convolutions in the diffusion backbone.

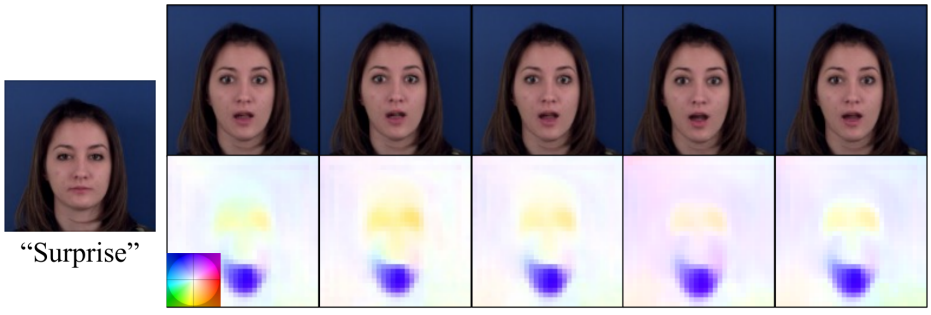


Figure 2.5: Examples of generated video frames and latent flow sequences produced by LFDM. The first column shows the given image x_0 and condition y. The latent flow maps represent backward optical flow to x_0 in the latent space. Flow is visualized using the color-coding scheme of Baker et al. [5]. Figure taken from the original LFDM paper [20].

2.4 Overview of Video Generation Techniques

Video generation can be viewed as the task of generating a sequence of images. A video is essentially an ordered set of frames that must be consistent both spatially and temporally. Spatial consistency ensures that each frame is visually coherent and realistic on its own, while temporal consistency guarantees smooth transitions between consecutive frames, preserving motion continuity and dynamic behavior across time. Most video generation techniques start from image generation methods and extend them to handle the temporal dimension.

2.4.1 GAN-based Approaches

The adversarial framework at the core of VICEGAN (Section 2.2) relies on Generative Adversarial Networks, a class of generative models originally introduced by Goodfellow et al. [12] for realistic static image synthesis. Having already seen their application in a concrete video-generation system, this section provides a more formal and general treatment of the GAN formulation and its extension to video.

A standard GAN consists of two neural networks trained in an adversarial way: a generator G and a discriminator D. The generator maps a random noise vector $z \sim p(z)$ to a synthetic image $x = G(z)$, while the discriminator aims to distinguish between real images $x \sim p_{\text{data}}(x)$ and generated ones. The training objective is formulated as a minimax game:

$$\min_G \max_D \; \mathbb{E}_{x \sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z)))]. \tag{2.3}$$

Through this process, the generator learns to produce images that are visually realistic and spatially coherent [12].

Extending GANs from image generation to video generation introduces the additional challenge of modeling temporal consistency. While each video frame must remain spatially realistic, consecutive frames must also exhibit smooth and coherent transitions over time. Formally, a video can be represented as a sequence of frames $\{x_1, x_2, \ldots, x_T\}$, where the joint distribution must capture both spatial and temporal dependencies:

$$p(x_1, x_2, \ldots, x_T) \neq \prod_{t=1}^{T} p(x_t). \tag{2.4}$$

GAN-based video generation methods address this issue by explicitly incorporating the temporal dimension into the generative process. One common approach is to generate an entire video clip as a spatio-temporal volume using 3D convolutional architectures, where the generator maps a latent vector z to a sequence of frames $V = G(z) \in \mathbb{R}^{T \times H \times W \times C}$. These models are able to learn joint spatial and temporal features directly.

Another line of work decomposes the latent space into separate components for appearance and motion, for example by sampling a static content vector z_c and a time-dependent motion vector z_t, and generating each frame as:

$$x_t = G(z_c, z_t), \quad t = 1, \ldots, T. \tag{2.5}$$

Additionally, temporal discriminators are often introduced to evaluate sequences of frames rather than individual images, enforcing temporal coherence by penalizing unrealistic transitions.

2.4.2 Autoregressive Models

According to the previous description, video data can be modeled as an ordered sequence of frames. In autoregressive formulations, each frame is generated conditionally on the previously observed frames [28].

$$p(x) = \prod_{i=1}^{N} p(x_i | x_0, x_1, x_2, \ldots, x_{i-1}; \theta), \tag{2.6}$$

where $p(x_i | x_0, x_1, x_2, \ldots, x_{i-1}; \theta)$ is the conditional distribution for frame x_i, and θ denotes the model parameters.

Training is typically formulated as maximum likelihood estimation (MLE), i.e., maximizing the sequence log-likelihood. Equivalently, one minimizes the negative log-likelihood (NLL):

$$\mathcal{L}_{\text{NLL}}(\theta) = -\sum_{i=1}^{N} \log \left(p(x_i | x_0, x_1, x_2, \ldots, x_{i-1}; \theta) \right). \tag{2.7}$$

The autoregressive model [18] ensures that each frame is generated with high quality, while also exploiting the dependencies between frames to maintain the spatiotemporal consistency of the generated video.

During training, these models efficiently capture spatiotemporal dependencies typically through "teacher forcing," where the ground truth of previous frames is provided to predict the next one. This compels the model to learn the underlying causal structure and temporal dynamics of the video data. However, this paradigm introduces specific trade-offs. A primary advantage is the ability to explicitly model complex dependencies, often resulting in sequences with high temporal coherence and logical continuity. Conversely, a significant disadvantage is the inference latency; due to the sequential nature of generation, frames cannot be generated in parallel. Furthermore, they are prone to error propagation (often referred to as exposure bias), where minor artifacts in early generated frames accumulate, potentially leading to quality degradation or "drift" in longer sequences.

To enable controllable generation, these models employ various conditioning techniques. The probability distribution is conditioned on auxiliary information c, modifying the formulation to $p(x_t|x_{<t}, c)$. Techniques such as prepending conditioning tokens to the sequence, using cross-attention mechanisms (especially in Transformer-based backbones), or feature concatenation allow the integration of guidance signals like text prompts, initial frames, or semantic labels to steer the video generation process successfully.

2.4.3 Transformer-based Methods

Transformer-based methods were originally introduced for natural language processing, where they showed that long-range dependencies in a sequence can be modeled effectively through self-attention. Their success quickly transferred to the visual domain and they are now a fundamental tool for image and video understanding, often replacing or complementing convolutional and recurrent architectures.

Vision Transformers for Visual Representation

In vision, Transformers are typically applied by viewing an image as a sequence of patches. Instead of processing pixels with local convolutional filters, the model receives a set of patch embeddings and learns how each patch should interact with all the others. This patch-based formulation allows the network to reason globally about the scene and to connect distant regions already in the early layers of the architecture. When trained on large-scale datasets, Vision Transformers

(ViT) provide powerful image representations that have proved competitive with, and in many cases superior to, those of traditional convolutional networks [10].

In many generative models, these visual Transformers are used primarily as encoders. Given an input frame or a reference image, the Transformer extracts a compact set of tokens that summarize high-level content and layout. These tokens can then be used as conditioning information for downstream components such as GAN generators, diffusion denoisers, or sequence models over latent codes. In this way, Transformers act as a flexible interface between raw visual data and the generative mechanism.

Transformers for Video Generation

Extending Transformers from images to videos requires taking the temporal dimension into account. A video can be seen as a sequence of frames, and each frame can again be decomposed into patches. Video Transformers process the resulting space-time tokens by combining spatial and temporal attention. Some architectures attend jointly over all tokens, while others adopt factorized schemes that alternate spatial and temporal attention blocks to keep the computational cost manageable.

This design gives the model direct access to information coming from distant frames, making it easier to capture long-term motion patterns and to maintain consistency over time. In practice, Video Transformers are employed in several ways within generative pipelines. Autoregressive models use them to predict the next latent token in a video sequence, after compressing frames into a discrete representation with a learned encoder. Other approaches use a Transformer as a conditioning backbone: it encodes text prompts, reference images, or short input clips and provides context to a generator that produces the final video. This is the case, for example, in many recent text-to-video and image-to-video systems.

Finally, Transformer blocks are increasingly integrated directly inside diffusion models for video. Instead of relying only on 3D convolutions, the denoising network is augmented with spatio-temporal attention layers that allow the model to look at all frames when predicting the noise at a given timestep. This tends to reduce temporal artifacts such as flickering and identity drift, and is particularly beneficial when videos are long or when fine-grained temporal control is required, as in controllable facial expression generation.

2.4.4 Diffusion Models for Image and Video Generation

Diffusion models are an effective class of deep learning-based generative models that challenge the dominance of GANs in generation tasks. Unlike GANs, which directly transform noise into data, diffusion models learn to reverse a gradual degradation process. This approach allows for more stable training, produces high-quality samples, and better captures the diversity of the data, making them particularly suitable for complex generation tasks.

2.4.5 Fundamentals of Diffusion Models

The theoretical foundation of diffusion models is based on a probabilistic framework that describes data transformation through gradual noise addition. The generative process is defined as a parameterized Markov chain that reverses this diffusion process. This framework consists of two distinct processes: a fixed forward process (or diffusion process) that gradually adds Gaussian noise to the data until the signal is destroyed, and a learnable reverse process that reconstructs the data structure from the noise. By learning the gradients of the data distribution (score matching) or estimating the noise added at each step, the model can iteratively refine a random noise vector into a coherent sample from the target distribution.

2.4.6 Denoising Diffusion Probabilistic Models (DDPM)

Denoising Diffusion Probabilistic Models (DDPM), introduced by Ho et al. [15], are a specific instantiation of the diffusion framework that achieves high-fidelity image generation. In DDPMs, the forward process is a fixed Markov chain that gradually adds Gaussian noise to the data according to a predefined variance schedule. This can be formally written as:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}), \tag{2.8}$$

where β_t defines the variance schedule and $t = 1, \ldots, T$ indexes the diffusion steps.

The reverse process is also modeled as a Markov chain with learned Gaussian transitions:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)), \tag{2.9}$$

where $\boldsymbol{\mu}_\theta$ and Σ_θ are predicted by a neural network parameterized by θ.

The key innovation of DDPM is the connection between the diffusion formulation and denoising score matching. Specifically, the training objective can be simplified to a weighted mean squared error between the true noise added to the image and the noise predicted by the network:

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}, t}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|^2\right], \tag{2.10}$$

where $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, and $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$.

This formulation enables stable training and high-quality synthesis, allowing the model to gradually denoise a random Gaussian sample into a coherent image from the target distribution.

2.4.7 Conditional Diffusion Models

Conditional diffusion models extend the standard diffusion framework to generate samples that satisfy specific conditions, such as class labels, textual descriptions, or other modalities. In this setting, both the forward and reverse processes are conditioned on additional information c, which guides the generative process towards samples consistent with the condition.

The reverse process is thus modeled as:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, c) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t, c), \Sigma_\theta(\mathbf{x}_t, t, c)). \tag{2.11}$$

Classifier Guidance One approach to conditional generation is *classifier guidance*, where an external classifier $p_\phi(y \mid \mathbf{x}_t)$ is used to steer the reverse diffusion process. During sampling, the predicted noise mean is adjusted by the gradient of the log-probability of the desired class with respect to the current sample:

$$\boldsymbol{\mu}_\theta^{\text{guided}}(\mathbf{x}_t, t, c) = \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) + s \cdot \Sigma_\theta(\mathbf{x}_t, t)\nabla_{\mathbf{x}_t} \log p_\phi(c \mid \mathbf{x}_t), \tag{2.12}$$

where s is a scaling factor controlling the strength of the guidance. This method can generate high-quality samples according to the condition, but it requires

training a separate classifier. [9]

Classifier-Free Guidance Another used technique is *classifier-free guidance*, which integrates conditioning directly into the diffusion model, without the need of a separate classifier. During training, the model is randomly provided with either the condition c or a null token (effectively unconditioned) with a fixed probability p_{drop}. The network thus learns both conditional and unconditional denoising:

$$\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \tilde{c}) \quad \text{where} \quad \tilde{c} = \begin{cases} c, & \text{with probability } 1 - p_{\text{drop}} \\ \varnothing, & \text{with probability } p_{\text{drop}} \end{cases}. \tag{2.13}$$

At sampling time, conditional generation is achieved by combining the conditional and unconditional predictions:

$$\boldsymbol{\epsilon}_\theta^{\text{cfg}}(\mathbf{x}_t, t, c) = \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \varnothing) + gf \cdot \big(\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, c) - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \varnothing)\big), \tag{2.14}$$

where $gf > 1$ is the guidance scale controlling the strength of conditioning. This approach allows strong conditioning without training a separate classifier, is simple to implement, and has become the standard in many state-of-the-art conditional diffusion models such as text-to-image generators. [16]

2.4.8 Diffusion Models for Video Synthesis

Diffusion models can be extended from images to videos by treating a video as a spatio-temporal signal and training the denoising network to jointly model spatial structure and temporal evolution. A direct approach is to operate in pixel space and adopt architectures that process multiple frames together (e.g., via 3D convolutions or factorized spatio-temporal attention), so that the denoiser can exploit context from neighboring frames to reduce temporal artifacts such as flickering and identity drift. A representative example is the Video Diffusion Models framework by Ho et al. [17], which proposes a diffusion model tailored to video data and studies strategies to scale to longer and higher-resolution generations through conditional sampling schemes.

From the perspective of controllable facial video synthesis, the key takeaway is that diffusion-based pipelines offer a flexible way to inject conditioning signals

(e.g., reference identity frames, emotion labels, or text prompts) while maintaining global coherence, at the cost of increased computational demands compared to single-pass GAN generators.

2.5 Disentangled Representation Learning

Disentangled representation learning aims to separate the underlying factors of variation in the data into distinct, independent dimensions of a representation. In the context of facial analysis and generation, this typically involves separating identity features (invariant characteristics of a person) from expression features (transient deformations of the face), as well as other factors like pose or lighting. Achieving such disentanglement is crucial for controllable generation, allowing the manipulation of specific attributes (e.g., changing the emotion) while preserving others (e.g., identity).

2.5.1 FECNet: A Compact Embedding for Facial Expression Similarity

Traditional approaches to facial expression analysis often rely on discrete categorical emotions (such as the six basic emotions, as in Ekman's basic-emotion taxonomy [11]) or the Facial Action Coding System (FACS). However, facial expressions are inherently continuous and subtle, often defying rigid categorization. FECNet (Facial Expression Comparison Network), proposed by Vemulapalli and Agarwala [27], addresses this limitation by learning a compact, continuous embedding space that aligns with human visual perception of expression similarity.

The core idea is to map facial images into a low-dimensional Euclidean space (e.g., 16 dimensions) where the distance between two points corresponds to the perceptual dissimilarity between their expressions. To achieve this, the model is trained on a large-scale dataset, the Facial Expression Comparison (FEC) dataset, which contains triplets of images. Human raters annotated these triplets by selecting which two faces in a set of three were the most similar in terms of expression.

The network is trained using a triplet loss function, which encourages the embedding distance between the visually similar pair to be smaller than the distance to the third, dissimilar image. This approach allows FECNet to capture fine-grained

variations and intensity changes that categorical labels miss.

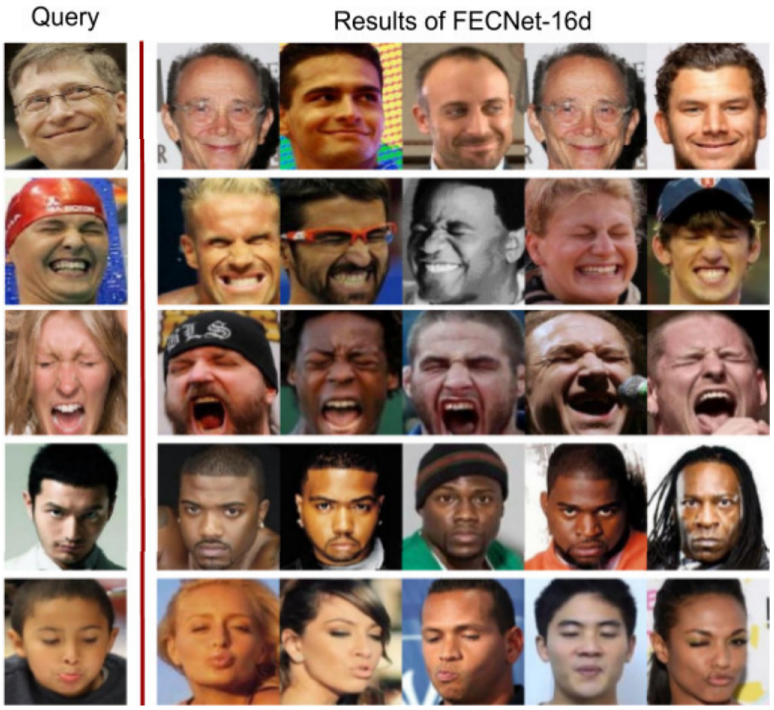


Figure 2.6: Top-5 retrieved images from the test set based on embedding similarity to the query. The figure demonstrates that the learned FECNet representations effectively preserve semantic and visual consistency, retrieving faces with similar expressions.

In the context of our video generation framework, FECNet will be used as a replacement for traditional motion codes. By leveraging the semantic structure of the embeddings generated by FECNet, we aim to guide the generative process with more meaningful and perceptually relevant motion representations. This approach is expected to enhance both the expressiveness and temporal coherence of the generated facial expressions. The specific integration and methodology will be detailed in Chapter 3.

2.6 MLP-Mixer

The MLP-Mixer architecture, introduced by Tolstikhin et al. [26], demonstrates that purely multi-layer perceptron (MLP)-based models can achieve competitive performance without relying on convolutional or attention mechanisms. The architecture operates on sequences of image patches treated as tokens and applies two types of MLP layers: *token-mixing MLPs* that enable communication be-

tween different spatial locations, and *channel-mixing MLPs* that enable communication between different feature channels. This design principle is relevant for video generation tasks, where modeling both spatial and temporal dependencies is crucial.

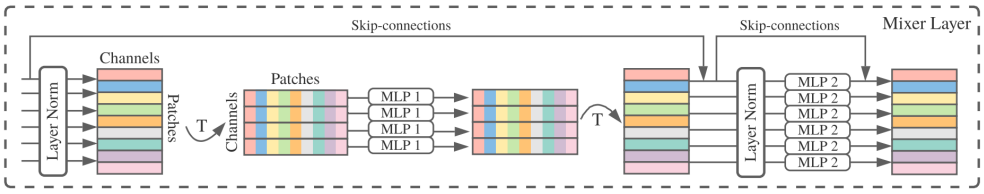


Figure 2.7: The architecture of MLP-Mixer. Figure from [26].

In the context of conditional video generation, MLP-Mixer offers several advantages for learning spatio-temporal dynamics. The token-mixing operation naturally extends to handle temporal sequences by treating video frames as tokens, allowing the architecture to capture motion patterns and temporal coherence through simple learned linear transformations. When integrated into conditioning mechanisms, MLP-Mixer can efficiently process conditioning information (such as emotion labels or motion codes) and propagate it across both spatial locations within frames and temporal positions across the sequence. This makes it particularly suitable for tasks requiring fine-grained control over generated content, as it can learn to modulate features based on conditioning signals while maintaining global consistency.

Compared to self-attention mechanisms commonly used in Transformer architectures, MLP-Mixer offers distinct computational advantages. While self-attention has quadratic complexity with respect to the number of tokens ($\mathcal{O}(N^2)$), MLP-Mixer maintains linear complexity ($\mathcal{O}(N)$), making it more scalable for high-resolution or long video sequences. Additionally, MLP-Mixer's architecture is conceptually simpler—it relies solely on matrix multiplications and nonlinearities without the need for specialized attention operations—which can lead to more stable training and easier implementation. The fixed-weight token-mixing approach also provides implicit inductive biases that can be beneficial for structured data like videos, where spatial and temporal relationships follow regular patterns. Although simple in design, MLP-Mixer has demonstrated that architectural sophistication (such as attention mechanisms) is not always necessary to achieve strong performance, suggesting that well-designed MLP architectures can serve as efficient and effective alternatives for processing sequential visual data in generative models.

Chapter 3

Methodology

This chapter presents in detail each architectural proposal introduced in this thesis for emotion-conditioned video generation. For each component, we analyze the expected advantages and explain the design choices that motivate its adoption. The effectiveness of each proposed modification will be evaluated through an ablation study, presented in Chapter 5, which quantifies the individual contribution of each method to the overall system performance. This systematic evaluation approach allows us to isolate and measure the impact of emotion-aware motion code generation, diffusion-based frame synthesis, and disentangled expression representations on the quality of generated emotional videos.

The analysis of VICEGAN's generated motion codes, presented in Section 2.2.3, revealed opportunities for enhancement in emotion-conditioned video generation. Two main directions are explored: first, conditioning the RNN on the target emotion to allow motion codes to carry semantic information about the emotion; second, replacing the GAN-based frame synthesis with diffusion models, which have shown improved training stability and generation quality in recent image synthesis work.

In this chapter, we present the different experimental settings proposed to develop this work, exploring various combinations of these enhancements to identify the most effective approach for emotion-conditioned video generation. For clarity, Table 3.1 summarizes the canonical names that will be used throughout the thesis to refer to the different model configurations.

Canonical name	Description
`baseline`	Original VICEGAN with GAN-based frame synthesis.
`onlyEM`	Conditioning shift applied to the motion generator, while keeping GAN-based frame synthesis.
`onlyDM`	Diffusion model (DM) replacing GAN-based frame synthesis.
`EM-DM`	Diffusion-based frame synthesis combined with the conditioning shift, with emotion injected in the motion generator.
`both_retrievedMcs`	Diffusion model driven by motion codes *retrieved* from real training sequences using an embedding network.
`LFDM`	State-of-the-art reference method.

Table 3.1: Canonical names and short descriptions of the model configurations considered in this thesis.

3.1 Setting onlyEM: Moving Emotion Conditioning

The first proposed methodology involves shifting the emotion conditioning from the frame synthesis stage to the motion code generator of VICEGAN. This modification is motivated by the observation that different emotions exhibit distinct temporal characteristics: for instance, surprise manifests rapidly and maintains peak intensity, while sadness develops gradually through subtle facial movements. By conditioning the recurrent dynamics on the target emotion, the generated motion codes are, in practice, *forced* to carry semantic information about the expression. In fact, in the proposed setting the frame generator no longer receives an explicit emotion conditioning signal; therefore, it must extract all the information needed to synthesize the correct expression solely from the motion codes. This constraint encourages the motion representation to encode not only generic facial dynamics, but also how the target expression naturally evolves over time.

This architectural shift offers several potential advantages. First, it allows the motion codes to inherently represent emotion-specific temporal patterns. Second, the separation of emotion conditioning (within motion generation) from content encoding (within the frame generator) improves the disentanglement between identity (*what* to generate) and its dynamic evolution (*how* it moves). Finally, the frame generator receives codes that are already emotion-informed, reducing the

complexity of the mapping from latent space to pixels. Instead of inferring both temporal dynamics and emotional expression from separate inputs, the generator focuses on synthesis using pre-conditioned codes, making the training process more stable and the resulting videos more realistic.

3.1.1 Network Architecture

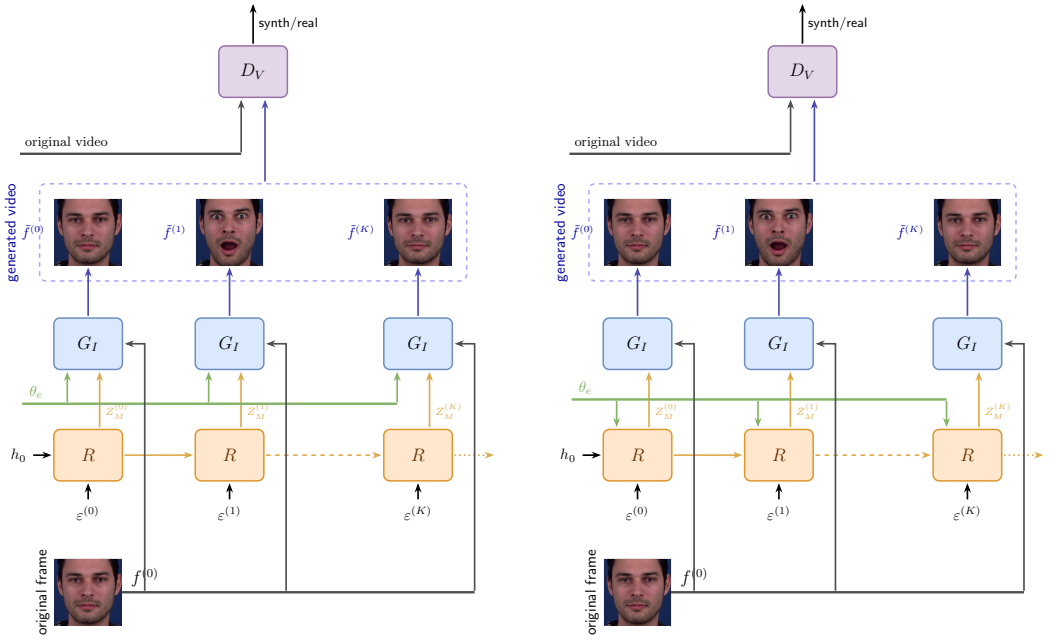
The implementation of this methodology focuses on restructuring the role of the category latent vector $\mathbf{z}_a \in \mathbb{R}^7$. In the original VICEGAN, $\mathbf{z}_a$ is treated as a static attribute, concatenated with the content $\mathbf{z}_c$ and motion $\mathbf{z}_m$ codes at the frame synthesis stage and applied uniformly across the sequence. In contrast, our proposed architecture moves this conditioning into the recurrent neural network (RNN) responsible for generating the motion codes.

The key modification concerns the GRU cell used for motion generation. While the baseline uses a simple GRU with an input dimension equal to `dim_z_motion` (10), the conditioned version expands this to `dim_z_motion + (dim_z_motion` $\times$ `dim_z_category)`, where `dim_z_category = 7` represents the emotion classes. This expansion accommodates the additional emotion-dependent information through an outer product mechanism.

At each time step t, the conditioning mechanism operates by first computing an outer product between the one-hot emotion vector $\mathbf{z}_a \in \mathbb{R}^7$ and the previous hidden state $\mathbf{h}_{t-1} \in \mathbb{R}^{10}$, producing a conditioning matrix $\mathbf{C}_t \in \mathbb{R}^{7\times10}$. This matrix is then flattened into a vector $\mathbf{c}_t \in \mathbb{R}^{70}$ and concatenated with the noise vector $\mathbf{e}_t \in \mathbb{R}^{10}$ to form the complete GRU input $[\mathbf{e}_t; \mathbf{c}_t] \in \mathbb{R}^{80}$. The GRU cell then processes this combined input along with the previous hidden state to compute the new hidden state $\mathbf{h}_t = \mathrm{GRU}([\mathbf{e}_t; \mathbf{c}_t], \mathbf{h}_{t-1})$, which serves both as output (motion code) and as input for the next time step.

This outer product formulation allows the network to learn emotion-specific modulations of the motion dynamics. Unlike simple concatenation, the outer product creates interaction terms between each emotion category and each dimension of the hidden state, providing richer representational capacity for encoding how different emotions influence temporal patterns.

Consequently, the frame generator's forward pass is simplified. While the original implementation concatenates three inputs—content code, motion code, and category code—resulting in a latent vector of dimension `dim_z_content + dim_z_motion + dim_z_category`, the modified version only concatenates



(a) Original VICEGAN. (b) Moving emotion conditioning.

Figure 3.1: Comparison between (a) the original VICEGAN architecture, where emotion conditioning θ_e is applied at the frame generation stage, and (b) the proposed approach, where θ_e is moved to the RNN to influence motion dynamics.

content and motion codes `dim_z_content + dim_z_motion`. The emotion information is now implicitly encoded in the motion codes $\mathbf{z}_m$ rather than being explicitly provided as a separate input.

3.1.2 Training Procedure

Since the only modification introduced in this setting is the relocation of emotion conditioning from the frame generator to the RNN, the training procedure and all hyperparameters are kept identical to the original VICEGAN baseline (described in detail in Section 2.2.2). This is a deliberate choice: by changing only the architecture and leaving the optimisation setup untouched, any difference observed in the results can be attributed only to to the architectural modification itself.

The training procedure maintains the adversarial framework of the original VICEGAN while adapting it to the new emotion-conditioned architecture. The system alternates between optimizing the video discriminator D_v and the generator G, following the standard GAN minimax game where the generator produces synthetic videos $\tilde{\mathbf{V}}$ using emotion-conditioned motion codes $\mathbf{z}_m$ and a frame synthe-

sizer F.

The same three-component loss structure is used for both the discriminator and the generator: adversarial loss, emotion classification loss, and identity classification loss, with weights $\lambda_{emo} = 1.0$ and $\lambda_{ide} = 1.0$.

$$\mathcal{L}(R, G_I, D_V) = \mathcal{L}_{adv} + \lambda_{emo}\,\mathcal{L}_{emo} + \lambda_{ide}\,\mathcal{L}_{ide}, \tag{3.1}$$

The training employs AdamW optimizers with learning rate $\eta = 0.0002$, momentum parameters $(\beta_1, \beta_2) = (0.5, 0.999)$, and batch size of 16 videos. Automatic Mixed Precision (AMP) is used to reduce memory consumption and accelerate training. The discriminator and generator are updated alternately with a 1:1 ratio to prevent either network from dominating the training dynamics.

3.2 Setting onlyDM: Diffusion-Based Frame Synthesis

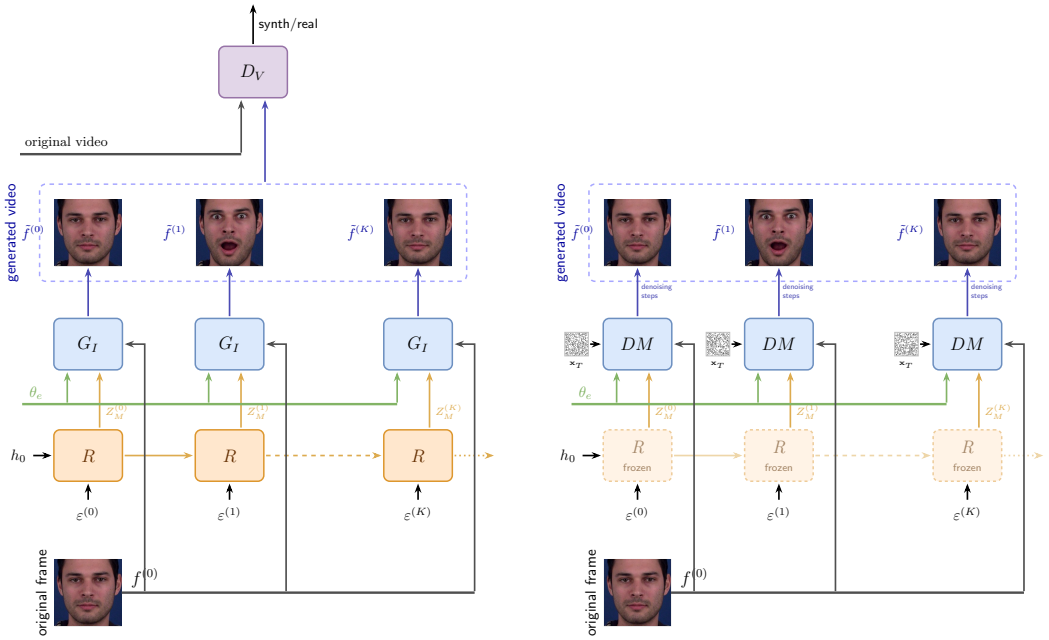
The second proposed methodology replaces VICEGAN's GAN-based frame synthesis with a diffusion model, while keeping the original motion code generation approach unchanged. This modification is motivated by recent progress in image synthesis: diffusion models often provide more stable optimization than adversarial training and can yield higher-quality samples.

In this setting, emotion conditioning remains applied at the frame generation stage, as in the original VICEGAN architecture; however, the diffusion model injects emotion through a conditioning mechanism integrated into the denoising process rather than via concatenation at the input of a GAN decoder. To isolate the contribution of diffusion-based synthesis, the pretrained VICEGAN motion code generator is kept frozen and used only to extract temporal motion codes during training and evaluation.

3.2.1 Network Architecture

The architectural modification introduces a U-Net-based denoising network [24] that replaces the original GAN decoder while preserving the motion code generation mechanism from VICEGAN. The system continues to use the recurrent neural network (GRU) to generate motion codes $\mathbf{z}_m^t \in \mathbb{R}^{10}$ for each time step t,

maintaining the same temporal dynamics as the original implementation. However, the frame synthesis process is fundamentally transformed from a single-pass generation to an iterative denoising procedure.



(a) Original VICEGAN. (b) Diffusion-based frame synthesis.

Figure 3.2: Comparison between (a) the original VICEGAN architecture and (b) the proposed diffusion-based approach, where the GAN decoder is replaced by a Diffusion Model (DM) while maintaining the frozen motion generator R.

The diffusion model operates on a forward diffusion process that gradually adds Gaussian noise to real frames over L timesteps ($L = 1000$), and a reverse denoising process that learns to reconstruct frames by iteratively removing noise. The noise schedule is defined by variance parameters $\beta_1, \ldots, \beta_L$ that increase linearly from $\beta_{\text{min}} = 0.0001$ to $\beta_{\text{max}} = 0.02$, controlling the amount of noise added at each diffusion step. From these, we compute cumulative products $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, which parameterize the forward process.

3.2.2 Diffusion Model Architecture

As anticipated, the diffusion model used in this thesis is based on a U-Net architecture adapted for our goal. It operates by progressively denoising a noisy input image $\mathbf{x}_T$ conditioned on the motion codes $\mathbf{z}_m$ and the identity reference $\mathbf{z}_c$. The conditioning is integrated into the network through dedicated injection mechanisms that modulate internal feature maps, rather than by concatenating

all conditioning variables at the input, as illustrated in Figure 3.3.

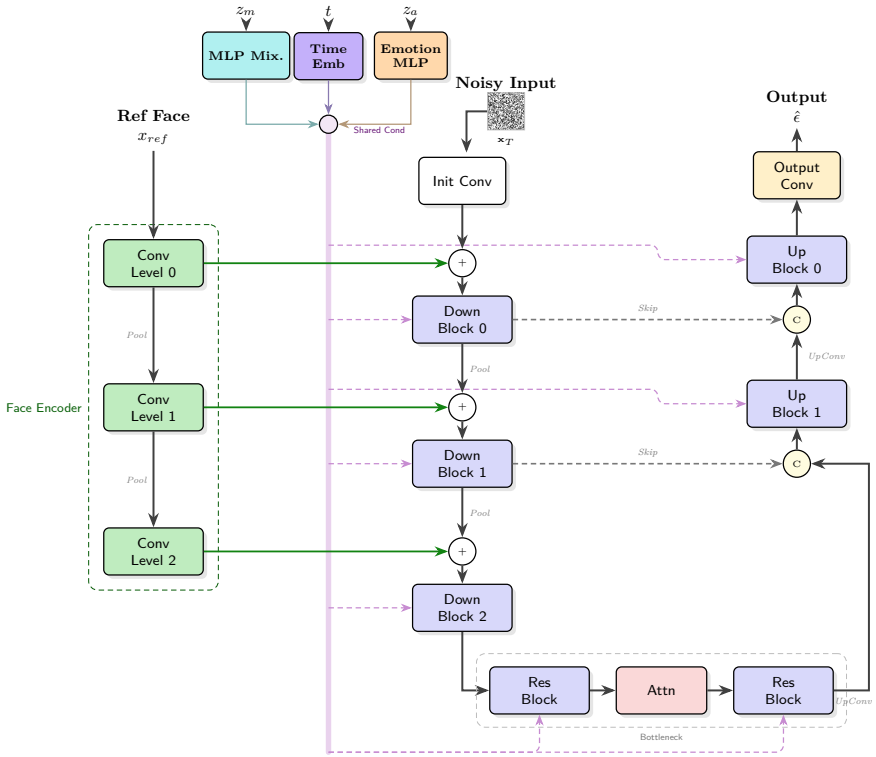


Figure 3.3: Detailed architecture of the conditional U-Net used in the diffusion-based settings. The network integrates identity features via spatial injection, motion codes through context-aware addition, and emotion labels using FiLM modulation to guide the denoising process.

The specific implementation of these conditioning mechanisms is designed to target different aspects of the generation process:

Identity Conditioning (Spatial Injection). To preserve the subject's appearance, a dedicated Face Encoder processes the neutral reference frame $\mathbf{x}_{ref}$ in parallel with the main denoising U-Net. The encoder extracts multi-scale feature maps, which are projected and added element-wise to the corresponding U-Net encoder features at each resolution level.

In this way, identity information, ranging from the overall facial structure to fine-grained skin details, remains available throughout the downsampling path, preventing the denoising process from generating a different identity.

Motion Conditioning (Context-Aware Addition). The motion information is provided as a sequence of latent codes $\mathbf{z}_m$. To ensure temporal smoothness, these codes are first processed by a *Temporal MLP Mixer* [26] that aggregates information from a local temporal window (context size $W = 7$). We refer to Section 2.6 for the background and general structure of the MLP-Mixer.

The resulting context-aware motion embedding is linearly projected and added to the feature maps within each residual block of the U-Net. This additive bias shifts the activation features, effectively guiding the network to deform the neutral face according to the driving motion dynamics without altering the semantic content.

Emotion Conditioning (FiLM). The categorical emotion label $\mathbf{z}_a$ is injected using *Feature-wise Linear Modulation* (FiLM) [23]. A dedicated MLP maps the emotion embedding to scale (γ) and shift (β) parameters.

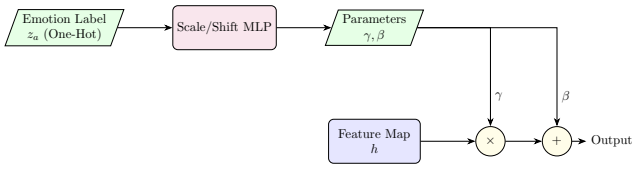


Figure 3.4: Emotion Conditioning Mechanism (FiLM). Emotion labels modulate feature statistics via learned affine transformations.

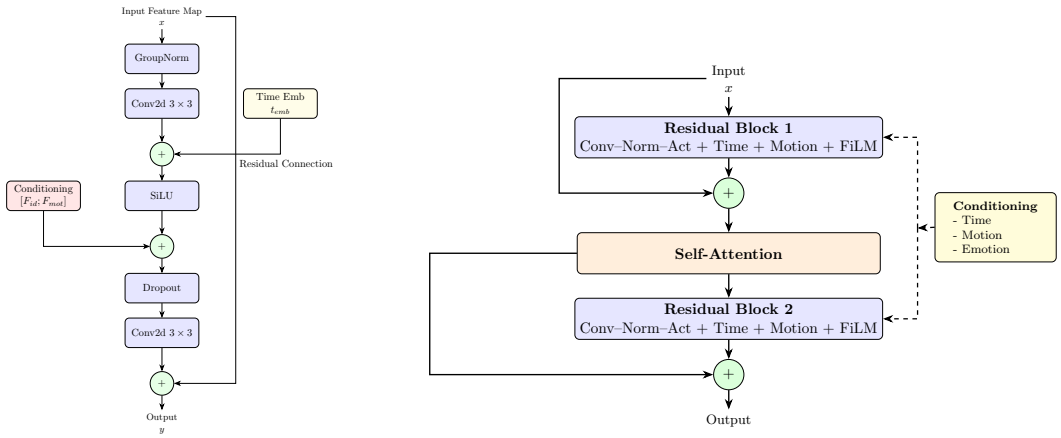
These parameters are used to modulate the normalized feature maps via an affine transformation: $\text{FiLM}(\mathbf{h}|\mathbf{z}_a) = \gamma(\mathbf{z}_a) \cdot \mathbf{h} + \beta(\mathbf{z}_a)$.

Time Conditioning. Finally, the diffusion timestep t is encoded using sinusoidal positional embeddings and added to the feature maps in every residual block, providing the network with the necessary signal to estimate the current noise level σ_t and perform the correct degree of denoising.

Component Details

To further clarify the internal structure of the key components, we provide detailed schematics of the U-Net building blocks and the MLP Mixer used for motion processing.

The MLP Mixer structure (see Section 2.6), tailored for processing vector sequences, replaces standard convolution or recurrent layers to mix information



(a) U-Net Residual Block (b) U-Net Bottleneck

Figure 3.5: Detailed structure of the U-Net components. (a) The Residual Block integrates time, identity, and motion conditioning at every level. (b) The Bottleneck processes the lowest resolution features with heavy conditioning.

across the temporal dimension (token mixing) and feature dimension (channel mixing), as shown below.

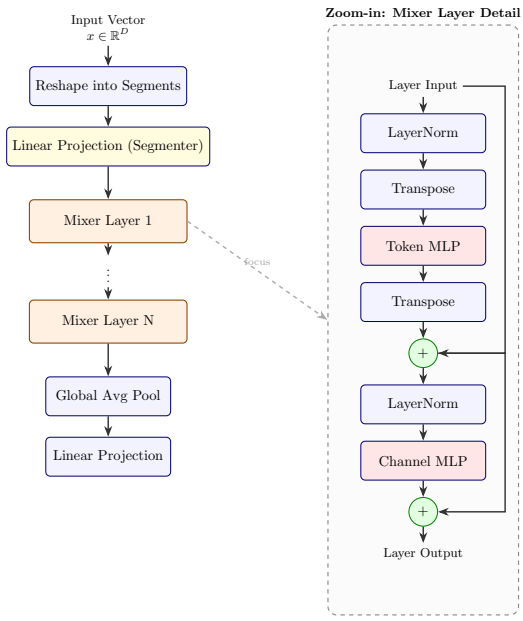


Figure 3.6: Architecture of the Vector MLP Mixer used for temporal motion smoothing. It alternates between mixing temporal information (Token MLP) and feature information (Channel MLP).

3.2.3 Training Procedure

The training procedure differs fundamentally from the adversarial framework of VICEGAN, replacing the minimax game between generator and discriminator with a regression objective that minimizes the error in noise prediction. The system is trained to denoise frames by learning the reverse diffusion process, conditioned on emotion and motion information.

For each training iteration, a batch of real videos $\mathbf{V} = \{\mathbf{x}_i\}_{i=1}^{T}$ and corresponding emotion labels y is sampled from the dataset. The first frame $\mathbf{x}_1$ is extracted as the neutral reference face, which serves as appearance conditioning throughout generation. Motion codes $\{\mathbf{z}_m^t\}_{t=1}^{T}$ are generated using the frozen pretrained motion encoder from the original VICEGAN, ensuring temporal consistency while isolating the evaluation of diffusion-based synthesis. The emotion label y is converted to a one-hot vector $\mathbf{z}_a \in \mathbb{R}^7$.

To implement classifier-free guidance the training randomly drops conditioning information with probability $p_{drop} = 0.1$. When a sample is selected for dropping, both motion and emotion codes are set to zero vectors, forcing the model to learn both conditional and unconditional generation. This enables flexible control over the conditioning strength at inference time through a guidance scale parameter.

The training processes frames either as complete videos or as random subsets, controlled by the `batch_mode` parameter. In `full_video` mode, all T frames are processed simultaneously, preserving temporal coherence. In `random_frames` mode, up to 8 randomly selected frames are processed, reducing memory consumption while maintaining coverage of the temporal span.

For each selected frame $\mathbf{x}$ with corresponding motion code $\mathbf{z}_m$ and emotion code $\mathbf{z}_a$, the training procedure:

1. Samples a random diffusion timestep $t \sim \mathcal{U}(0, L-1)$.
2. Samples Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ with the same dimensions as the frame.
3. Computes the noisy frame using the forward diffusion process:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x} + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon} \tag{3.2}$$

4. Predicts the noise using the U-Net: $\hat{\boldsymbol{\epsilon}} = \text{U-Net}(\mathbf{x}_t, \mathbf{x}_1, t, \mathbf{z}_m, \mathbf{z}_a)$.

5. Computes the mean squared error loss:

$$\mathcal{L}_{DDPM} = \|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}\|^2 \tag{3.3}$$

This simple regression objective trains the network to predict the noise component at any diffusion timestep, which can then be used to iteratively denoise random noise into a coherent frame during inference.

The optimization employs AdamW with learning rate $\eta = 0.0001$ (half that of the GAN-based generator to account for the more stable training dynamics), momentum parameters $(\beta_1, \beta_2) = (0.5, 0.999)$, and batch size of 16 videos. Automatic Mixed Precision (AMP) accelerates training and reduces memory consumption. Gradient clipping with maximum norm 1.0 prevents gradient explosion.

Unlike the alternating discriminator-generator updates in adversarial training, the diffusion model requires only generator updates at each iteration, simplifying the training loop and eliminating concerns about discriminator-generator balance. The stable regression objective typically results in smoother loss curves and more predictable convergence compared to the oscillatory dynamics often observed in GAN training.

3.3 Setting EM-DM: Combining Both Methodologies

The third setting combines the two previous proposals in a single pipeline. The overall procedure follows the same approach described in Setting 3.2: frame synthesis is performed by a diffusion model, and the evaluation focuses on the quality and stability benefits of diffusion-based generation.

The key difference lies in the source of the motion codes. Instead of using the original VICEGAN motion generator, we employ the emotion-conditioned motion code generator introduced in Setting 3.1 (RNN and Frame Synthesiser trained with explicit emotion conditioning in the recurrent dynamics).

This design allows us to jointly leverage (i) motion representations that are semantically meaningful with respect to the target expression and its temporal evolution, and (ii) the improved sample quality and optimization stability of diffusion-based synthesis. All remaining architectural choices and training details are kept consistent with Setting 3.2.

3.4 Setting both_retrievedMcs: Combining Both Methodologies in a Decoupled Architecture

The fourth setting integrates the two previous innovations into a unified pipeline that separately addresses dynamics generation (motion) and frame synthesis. The approach is decoupled since motion dynamics are provided by a **pretrained motion-code extractor** (FECNet), while frame synthesis is performed by the same diffusion-based synthesizer introduced in Setting 3.2. FECNet remains **frozen** and is used only to extract motion-code trajectories.

3.4.1 Specialized Preprocessing for FECNet

To ensure good performance of **FECNet** as a semantic feature extractor, we define a clear preprocessing pipeline for input video frames. This stage is important to ensure that extracted motion codes capture facial dynamics rather than changes in head pose or lighting. The processing begins by reading video frames directly from the original source archives and converting them into a standardized RGB representation. To meet the requirements of open-source computer vision libraries used for alignment, frames are temporarily transformed into the BGR color space. A core component of this pipeline is face alignment and normalization. Each frame is processed using *dlib* landmarks to align the subject's face based on eye position. Specifically, the face is resized to a fixed inter-ocular distance of 55 pixels and subjected to a 224×224 crop centered on the eyes. After this spatial normalization, the frames are returned to the RGB space, converted to floating-point tensors, and normalized using standard ImageNet statistics. This preprocessing ensures that the motion-code extractor receives standardized data and produces stable, accurate expression embeddings.

Figure 3.7: Facial preprocessing pipeline for FECNet. Left: Examples of raw input frames from the dataset. Right: The corresponding frames after Preprocessing pipeline for FECNet.

To further evaluate the impact of this specialized alignment, Figure 3.8 illustrates the difference between the FECNet-aligned frames and a standard resizing approach. By visualizing the detected eye centers and their relative distances, we can observe how the proposed preprocessing enforces a constant inter-ocular distance and a fixed vertical offset for the eyes. This geometric consistency is vital for the semantic feature extractor, as it ensures that variations in head pose or distance from the camera are not misinterpreted as facial expression dynamics.

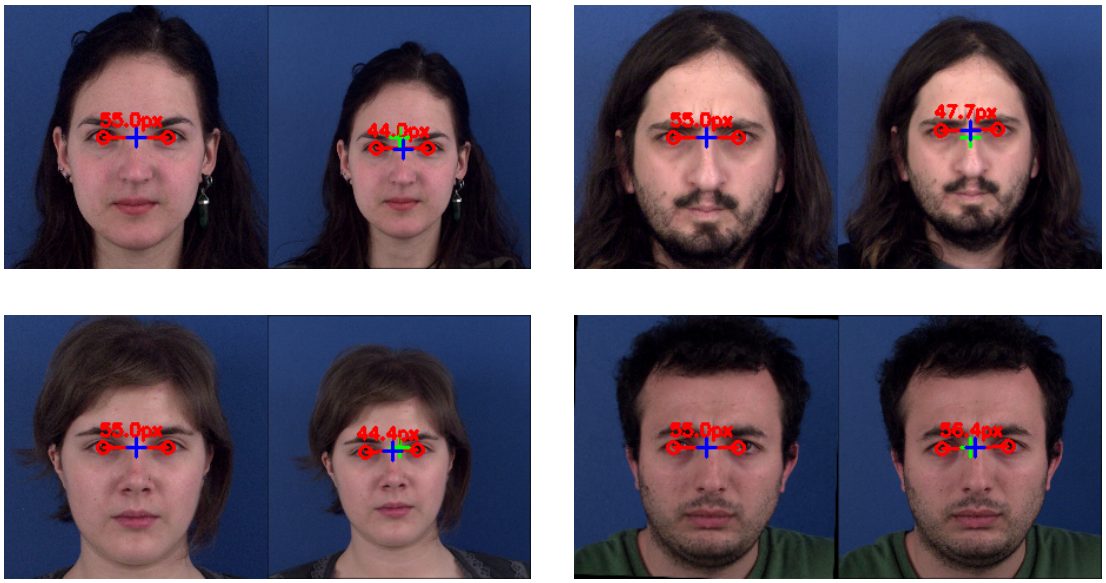


Figure 3.8: Comparison of eye alignment consistency. Each image shows the eye center markers and the inter-ocular distance line. The visual markers (red circles for eye centers, green for image center, blue for mid-eye point) demonstrate how FECNet alignment maintains a constant geometry across different subjects and frames, unlike a simple unaligned resize.

3.4.2 Network Architecture

The combined architecture for this setting organizes the generation process into two components that interact through a shared latent space. We use **FECNet** as an offline feature extractor and drive the diffusion model using **retrieved motion-code trajectories** taken from real training sequences. Given a target emotion, we select a real training video with the same emotion and extract its full sequence of FECNet motion codes. These codes provide a meaningful representation of expression dynamics. Following Setting 3.2, the diffusion-based synthesizer generates the final video frames from three inputs: the retrieved motion-code sequence, a neutral reference frame to preserve identity and random Gaussian noise $\mathbf{x}_T$ as the initial state for the denoising process.

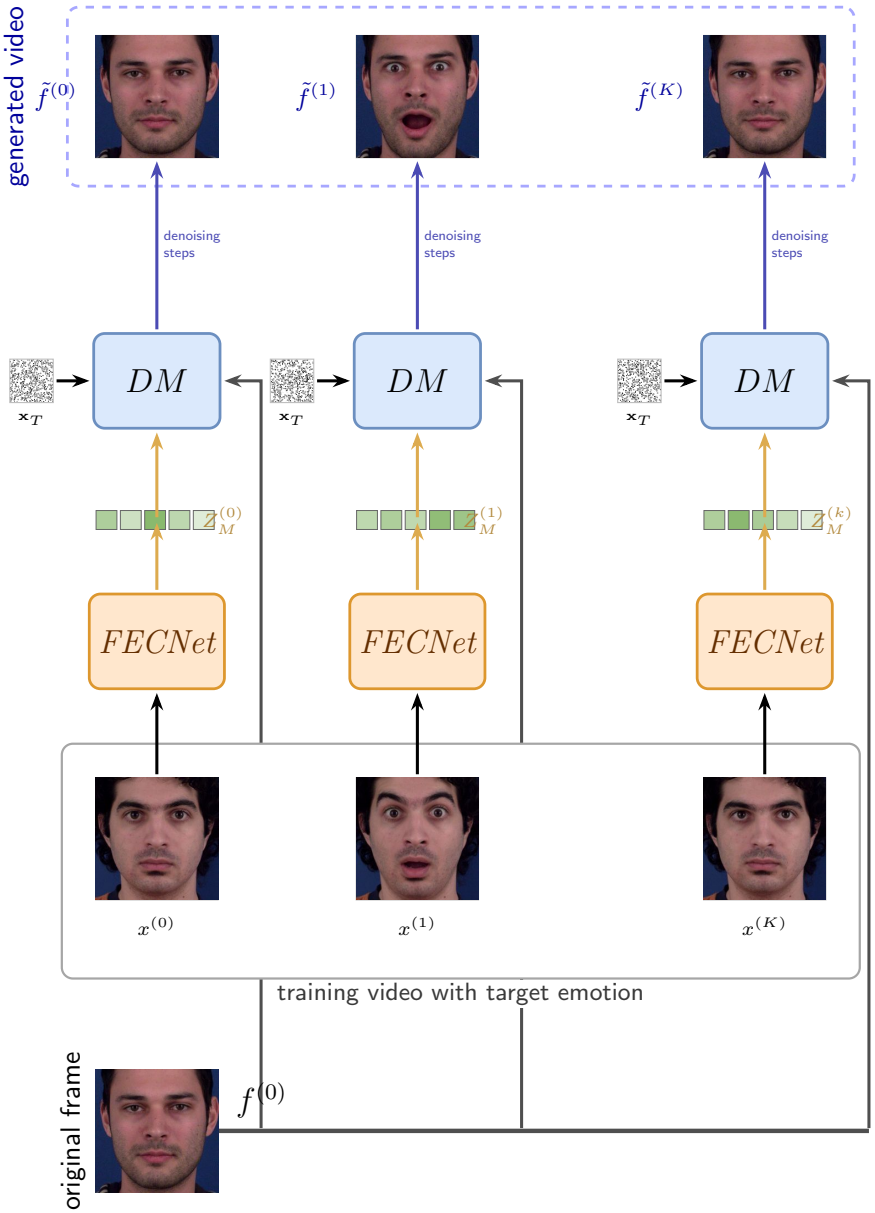


Figure 3.9: Combined architecture (Setting 4). Motion codes Z_M are retrieved from real training sequences using FECNet and, together with the identity reference $f^{(0)}$ and Gaussian noise $\mathbf{x}_T$, drive the Diffusion Model (DM) to synthesize the output sequence.

The diffusion model architecture (U-Net and conditioning mechanisms) is the same as described in Setting 3.2 (Section 3.2.2). In this setting, the only change is the *source* of motion codes $\mathbf{z}_m$: instead of using the frozen motion generator from VICEGAN, we use trajectories extracted from real training sequences with FECNet.

3.4.3 Diffusion-Based Frame Synthesizer Training

Unlike the original methodology that used motion codes learned unsupervised by VICEGAN, in this setting we use **FECNet** [27] as an offline feature extractor. For each real video in the training dataset, FECNet extracts a sequence of frame-by-frame embeddings that constitute the reference "motion codes" $\mathbf{z}_m^{real}$.

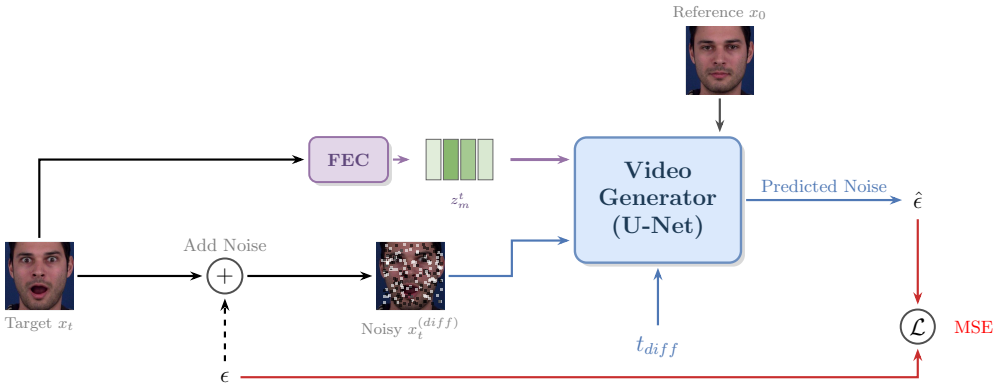


Figure 3.10: Training Diffusion Model. The Diffusion Model learns to reconstruct the target frame x_t starting from noise, conditioned on the motion code z_m^t extracted from the target itself by FECNet and the identity reference x_0.

The diffusion model (U-Net) is trained following the same procedure described in Setting 3.2; the key difference is the *source* of the motion-conditioning signal $\mathbf{z}_m$, which is now extracted by FECNet rather than produced by the VICEGAN motion generator (the RNN).

In practice, the expression-related information is already implicitly encoded in the FECNet embeddings, which are explicitly trained to discriminate facial expressions.

3.4.4 Motion Code Retrieval-Based Generation

In Setting 4, motion codes are provided through a retrieval strategy: we directly reuse motion codes extracted from real training sequences, while conditioning the synthesis on new identities not seen during training. This allows us to transfer emotion dynamics without training any motion-code generator. The generation pipeline is the following:

1. Given a neutral reference frame (identity) and a target emotion label, randomly select a real training video that exhibits the same target emotion.

2. Extract the full sequence of motion codes using FECNet (after the preprocessing step shown in Section 3.4.1).

3. Generate the output video by running the diffusion-based frame synthesizer conditioned on the neutral reference frame and on the retrieved motion code sequence.

Since the motion trajectory is taken directly from real data, no motion code generator is used at inference time, and the diffusion model is driven only by the retrieved motion codes (and the identity reference). During training, we also use motion mixing to better separate identity from facial dynamics. For each batch, the model first extracts motion codes from the input videos. Then, at each epoch, a mixing probability is increased linearly during a warmup phase up to a maximum value, and some samples are selected for mixing. For each selected sample, its motion code is replaced with one taken from a video with the same emotion but a different identity. This helps the model learn motion patterns that are less tied to a specific person and improves generalization.

Chapter 4

Experimental Setup

4.1 Introduction

This chapter describes the experimental protocol used to evaluate the proposed architectural enhancements. In particular, we summarize the dataset and preprocessing choices, the implementation setup used for training and evaluation, and the quantitative metrics employed throughout the analysis.

Quantitative and qualitative results, together with the user study, are reported in Chapter 5.

The remainder of this chapter is organized as follows:

- Section 4.2 introduces the MUG dataset and the adopted preprocessing and split.
- Section 4.3 summarizes the implementation and compute environment.
- Section 4.4 defines the metrics used to evaluate video quality, emotion accuracy, and identity/temporal consistency.

4.2 Dataset

In line with LFDM, and following the VICEGAN baseline protocol, we use the MUG Facial Expression Database [2] for both training and evaluation.

The MUG database was created by the Multimedia Understanding Group to provide a high-quality resource for facial expression analysis, overcoming limitations

of previous datasets regarding resolution and lighting uniformity. The complete database consists of image sequences from 86 subjects (35 women and 51 men) of Caucasian origin, aged between 20 and 35 years. Recordings were performed in a controlled environment with a blue screen background and uniform lighting provided by two 300W sources. The videos were captured at 19 frames per second with a high resolution of 896×896 pixels.

The dataset is organized into two parts: posed expressions and laboratory-induced emotions. In this thesis, we use the posed-expression subset, where subjects perform the six basic facial expressions (anger, disgust, fear, happiness, sadness, and surprise), as commonly framed in Ekman's basic-emotion taxonomy [11]. Each sequence follows a consistent temporal structure, starting and ending with a neutral expression and reaching an emotional peak (apex) in between.

Out of the 86 subjects, 52 are publicly available for research purposes; this subset is the basis of our experimental setup.

Video Structure

The videos in the MUG facial expression dataset follow a structured temporal pattern divided into three phases:

- **Frames 0-9 (10 frames)**: Initial neutral expression
- **Frames 10-21 (12 frames)**: Target emotion expression (apex phase)
- **Frames 22-31 (10 frames)**: Return to neutral expression

This structure reflects the natural dynamics of facial expressions, which typically transition from a neutral state to an emotional peak before returning to neutrality.

4.2.1 Preprocessing and Train-Test Split

The dataset is partitioned by subject identity to ensure that training and evaluation identities are disjoint. The test set includes 10 identities (subjects 17, 34, 45, 55, 60, 64, 65, 73, 78, and 83), while the training set includes the remaining 42 identities.

For each video, face detection and alignment are performed using a facial landmark detector, with crops normalized to a standard 256×256 resolution. From

this aligned image, we generate two resized versions depending on the downstream component: 224×224 pixels for feature extraction (e.g., FECNet-based motion codes; Section 3.4.1), and 64×64 pixels for the generative models. Each clip is temporally subsampled to exactly 32 frames using uniform sampling to maintain consistent temporal coverage. Pixel intensities are normalized to $[-1, 1]$ using $(x - 0.5)/0.5$, with $x \in [0, 1]$.

We apply the following filtering criteria to both subsets: (i) only clips with at least 32 frames are retained; (ii) sequences labeled as "mixed" or "extra" are excluded to focus on the seven primary categories (anger, disgust, fear, happiness, sadness, surprise, and neutral); and (iii) clips that do not terminate with a neutral expression are discarded to ensure a consistent sequence structure.

The final composition of the subsets is reported in Table 4.1. The training set consists of 761 sequences across 42 identities (24,352 sampled frames), while the test set includes 158 sequences from 10 identities.

	Training set		**Test set**	
Emotion	**Videos**	**Frames**	**Videos**	**Frames**
Anger	133	4,256	30	960
Disgust	121	3,872	30	960
Fear	108	3,456	17	544
Happiness	135	4,320	31	992
Sadness	112	3,584	21	672
Surprise	138	4,416	29	928
Total	**761**	**24,352**	**158**	**5,056**

Table 4.1: Composition of training and test sets by emotion category.

The number of sequences per identity in the training set ranges from 10 (subject 076) to 32 (subject 002), with an average of approximately 18 videos per identity, reflecting the natural variability of the MUG database.

4.3 Implementation Details

All experiments were conducted on the high-performance computing cluster of the University of Twente. Training uses a multi-GPU setup with 3 NVIDIA L40 GPUs (48 GB VRAM each). For inference and evaluation, a single L40 GPU is used to generate samples and compute the metrics defined in Section 4.4.

The generative models and most evaluation components are implemented in PyTorch (v2.0.1) with CUDA 11.7. The Fréchet Video Distance (FVD) computation

relies on the original TensorFlow-based reference implementation (Section 4.3.1).

4.3.1 Tools, Technologies, and Models Used for Implementation

The evaluation pipeline for assessing the quality of generated facial expression videos relies on a set of specialized deep learning models, software libraries, and computational frameworks. This section describes each tool employed and its specific role in the evaluation process.

Deep Learning Frameworks

PyTorch PyTorch [22] serves as the primary deep learning framework for the entire video generation and evaluation system. It is used for loading and running the generative models (VICEGAN variants, DDPM-based generators, and the Latent Flow Diffusion Model (LFDM) baseline [20]), performing data preprocessing through `torchvision.transforms`, and managing GPU-accelerated inference via `torch.cuda` and `nn.DataParallel` for multi-GPU support. All generative models are implemented as PyTorch `nn.Module` subclasses, and checkpoints are serialized in PyTorch's native `.pth` format.

TensorFlow TensorFlow 1.x (via `tensorflow.compat.v1`) [1] is employed exclusively for the computation of the Fréchet Video Distance (FVD). The FVD pipeline uses the original Google Research reference implementation[1], which requires TensorFlow for loading the pre-trained I3D model from TensorFlow Hub and computing the Fréchet distance through the `tensorflow_gan` (TF-GAN) library. A dedicated TensorFlow virtual environment is maintained separately from the PyTorch environment to avoid dependency conflicts.

Feature Extraction Models

I3D (Inflated 3D ConvNet) — FVD Computation The Inflated 3D Convolutional Network (I3D) [6], pre-trained on the Kinetics-400 action recognition dataset, is used as the backbone feature extractor for FVD computation. The model is loaded from TensorFlow Hub

[1] https://github.com/google-research/google-research/tree/master/frechet_video_distance

(`deepmind/i3d-kinetics-400/1`). Input videos—both real and generated—are resized to 224×224 and scaled to $[-1, 1]$ before being fed to the network. The extracted features are then used to compute the mean vector and covariance matrix for the Fréchet distance calculation, which is performed using `tfgan.eval.frechet_classifier_distance_from_activations`.

DeepFace — Emotion Recognition and Face Embeddings DeepFace [25] is a Python library for facial analysis that provides unified access to multiple face recognition and emotion analysis models. In our pipeline, DeepFace is used for:

- **Emotion recognition** to compute AEA/DEA (via `DeepFace.analyze(..., actions=[emotion])`).
- **Face embeddings** to compute ACD-I/ACD-C (via `DeepFace.represent()`) using VGG-Face [21], OpenFace [3], and ArcFace [8].

Face detection enforcement is disabled (`enforce_detection=False`) to improve robustness on low-resolution (64×64) generated frames.

Numerical and Scientific Libraries

NumPy NumPy [13] is used extensively for numerical operations on extracted feature vectors and embeddings, including mean and covariance computation.

Matplotlib Matplotlib [19] is used for generating diagnostic plots, including per-frame AEA and ACD-I profiles across the 32-frame video sequence and per-emotion breakdowns of the evaluation metrics.

Data Management and Processing

Dataset organization, preprocessing, and the train-test split are described in Section 4.2.1. For evaluation, we cache preprocessed clips to avoid repeated decoding and resizing when computing multiple metrics. To ensure fair comparisons across models, all generated frames are normalized to the same evaluation format before metric extraction. LFDM natively generates frames at 128×128, while our frameworks generate at 64×64. Therefore, we apply the same face-centered crop-and-resize procedure used in training preprocessing and convert all generated

frames to 64×64 before computing metrics. This step reduces resolution-related bias in the feature extractors of the models used in evaluation protocol.

Computational Infrastructure

The evaluation pipeline is designed to run on GPU-accelerated compute nodes managed through a SLURM workload manager. Separate SLURM job scripts are used for different evaluation phases: sample generation, identity representation extraction, emotion representation extraction, ACD computation, emotion metric computation, and FVD computation. The FVD computation requires a dedicated job using the TensorFlow virtual environment, while all other metrics run under the PyTorch environment. Multi-GPU inference is supported via PyTorch's `DataParallel` wrapper for the sample generation phase.

Summary of Tools

Table 4.2 provides a concise summary of the tools, models, and libraries employed in the evaluation pipeline.

Component	**Tool / Model**	**Purpose**
DL Framework	PyTorch	Model inference, preprocessing
DL Framework	TensorFlow 1.x	FVD computation (I3D + TF-GAN)
Feature Extractor	I3D (Kinetics-400)	Video-level features for FVD
Face Analysis	DeepFace	Emotion recognition, identity embeddings
Face Recognition	VGG-Face	Identity embeddings (ACD)
Face Recognition	OpenFace	Identity embeddings (ACD)
Face Recognition	ArcFace	Identity embeddings (ACD)
Scientific Comp.	NumPy	Statistics, Math operations
Visualization	Matplotlib	Diagnostic plots
Data	MUG Dataset	Facial expression videos
Infrastructure	SLURM + GPU nodes	Job scheduling, GPU acceleration

Table 4.2: Summary of tools and models used in the evaluation pipeline.

4.4 Evaluation Metrics for Video Generation

Evaluating the quality of generated videos is a complex task that requires assessing multiple aspects of the synthesis process. Unlike static image generation, video generation must be evaluated not only for visual realism and spatial coherence but also for temporal consistency, motion dynamics, and the preservation of semantic attributes across frames. In the context of facial expression video generation, additional challenges arise from the need to assess both the accuracy of emotional expressions and the preservation of facial identity throughout the sequence.

This section defines the metrics used to assess the performance of video generation models, with a focus on facial expression synthesis. The metrics can be grouped into three categories: (1) overall video quality and realism (FVD), (2) emotion expression accuracy (AEA and DEA), and (3) identity preservation and temporal consistency (ACD).

4.4.1 Fréchet Video Distance (FVD)

The Fréchet Video Distance is a metric designed to measure the quality and realism of generated videos by comparing their distribution with that of real videos from the dataset. FVD extends the concept of Fréchet Inception Distance (FID), originally developed for image evaluation, to the video domain by computing the same distributional distance on features extracted from an action-recognition backbone (e.g., I3D) applied to video clips.

Computation

The computation of FVD follows a structured pipeline:

In our experiments, FVD is computed using the original TensorFlow reference implementation from Google Research, which loads the I3D backbone from TensorFlow Hub (`deepmind/i3d-kinetics-400/1`) and computes the Fréchet distance using `tfgan.eval.frechet_classifier_distance_from_activations`.

Feature Extraction Both real and generated videos are processed through a pre-trained I3D (Inflated 3D ConvNet) model [6], originally trained on the Kinetics-400 action recognition dataset. The I3D architecture extends 2D Con-

vNets to 3D by inflating the convolutional filters and pooling kernels, enabling the model to capture both spatial and temporal features from video sequences. For each 32-frame clip, the I3D model extracts 1024-dimensional feature vectors from the global average pooling layer. For evaluation, both real and generated clips are resized to 224×224 before feature extraction.

In the evaluation code, both real and generated clips are loaded at 64×64 resolution and then upsampled to 224×224 via bilinear interpolation before being fed to I3D.

Statistical Computation For both the real and generated video sets, the mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ are computed from the extracted features:

$$\boldsymbol{\mu}_{\text{real}} = \frac{1}{N_{\text{real}}} \sum_{i=1}^{N_{\text{real}}} \mathbf{f}_i^{\text{real}}, \quad \boldsymbol{\Sigma}_{\text{real}} = \frac{1}{N_{\text{real}} - 1} \sum_{i=1}^{N_{\text{real}}} (\mathbf{f}_i^{\text{real}} - \boldsymbol{\mu}_{\text{real}})(\mathbf{f}_i^{\text{real}} - \boldsymbol{\mu}_{\text{real}})^\top \quad (4.1)$$

and similarly for the generated videos.

Fréchet Distance Calculation The FVD is then computed as the Fréchet distance (also known as Wasserstein-2 distance) between two multivariate Gaussian distributions:

$$\text{FVD} = \|\boldsymbol{\mu}_{\text{real}} - \boldsymbol{\mu}_{\text{gen}}\|_2^2 + \text{Tr}\left(\boldsymbol{\Sigma}_{\text{real}} + \boldsymbol{\Sigma}_{\text{gen}} - 2\sqrt{\boldsymbol{\Sigma}_{\text{real}}\boldsymbol{\Sigma}_{\text{gen}}}\right) \quad (4.2)$$

where $\|\cdot\|_2$ denotes the Euclidean norm, $\text{Tr}(\cdot)$ is the matrix trace, and $\sqrt{\boldsymbol{\Sigma}_{\text{real}}\boldsymbol{\Sigma}_{\text{gen}}}$ is the matrix square root of the product.

Interpretation

The FVD metric ranges from 0 to $+\infty$, with lower values indicating better quality:

- **FVD = 0**: The distribution of generated videos is identical to the distribution of real videos
- **Higher FVD**: Greater divergence between generated and real video distributions

FVD can be computed globally across all videos in the dataset or separately for each emotion category, providing insights into which emotions are synthesized

more realistically. The metric is valuable as a distribution-level proxy of realism in a learned feature space. However, it should not be interpreted as a direct measure of temporal smoothness: for temporal coherence we rely on explicit continuity metrics (ACD-C) and on frame-wise trends (e.g., AEA/ACD-I profiles).

4.4.2 Average Emotion Accuracy (AEA)

The Average Emotion Accuracy (AEA) metric evaluates how accurately generated videos express the target emotion specified during generation. This metric is used for assessing whether the model successfully learns to synthesize the intended emotional expressions.

Computation

The AEA computation involves the following steps:

Emotion Classification Each frame in the generated video is analyzed using DeepFace [25], a deep learning library for facial analysis. DeepFace provides automatic emotion recognition by classifying each frame into one of seven categories: the six basic emotions (angry, disgust, fear, happy, sad, surprise) plus neutral. For each frame t, the library returns the predicted dominant emotion $\hat{e}_t$.

Standard AEA (central frames) In this thesis, AEA is computed on the central frames only (the 12-frame apex window, 10–21), directly reflecting how well the model expresses the target emotion when it should be most prominent. Formally, for a video with target emotion e_{target}:

$$\text{AEA} = \frac{1}{N_{\text{videos}}} \sum_{v=1}^{N_{\text{videos}}} \frac{1}{12} \sum_{t=10}^{21} \mathbb{1}_{\{\hat{e}_{v,t}=e_{\text{target},v}\}} \tag{4.3}$$

where $\mathbb{1}_A$ is the indicator function of the event A.

AEA With Neutral (AEA_{WN}) AEA_{WN} extends the scoring to all 32 frames by assigning a per-segment target: neutral in the onset (frames 0–9) and offset (frames 22–31), and the target emotion in the apex (frames 10–21). It measures

the average per-frame accuracy under these segment-specific targets:

$$\text{AEA}_{WN} = \frac{1}{N_{\text{videos}}} \sum_{i=1}^{N_{\text{videos}}} \frac{1}{32} \Big(\sum_{t=0}^{9} \mathbf{1}(y_{i,t} = \text{neutral}) + \sum_{t=10}^{21} \mathbf{1}(y_{i,t} = \ell_{\text{target},i}) + \sum_{t=22}^{31} \mathbf{1}(y_{i,t} = \text{neutral}) \Big) \tag{4.4}$$

This metric rewards videos that transition correctly through all three phases (neutral → apex → neutral), not only those that produce the correct apex emotion.

Interpretation

AEA values range from 0 to 1, with higher values indicating better performance:

- **AEA**: Average classification accuracy on the central frames of generated videos
- $\mathbf{AEA_{WN}}$: Average per-frame accuracy over all 32 frames, where each frame is scored against a per-segment target (neutral in onset/offset, target emotion in apex)
- **AEA = 1.0**: Perfect accuracy, i.e., all central frames (10–21) match the target emotion
- $\mathbf{AEA_{WN} = 1.0}$: Perfect accuracy across all frames, with correct temporal structure (neutral in onset/offset, target emotion in apex)
- $\mathbf{AEA/AEA_{WN} = 0.0}$: No videos have any frames matching the target emotion (or segment-specific targets in AEA_{WN})

The AEA metric provides a continuous measure of emotional expression quality, making it sensitive to partial successes where some, but not all, frames display the correct emotion.

4.4.3 Dominant Emotion Accuracy (DEA)

While AEA measures average frame-level agreement, Dominant Emotion Accuracy (DEA) evaluates the rate of videos in which the most prominent recognized

emotion in the central frames is the correct one.

Computation

Majority-Agreement Criterion A video is considered correct under DEA if more than 50% of the central frames (10–21) match the target emotion:

$$\text{dominant}_v = \begin{cases} 1, & \text{if } \frac{1}{12}\sum_{t=10}^{21} \mathbb{1}_{\{\hat{e}_{v,t}=e_{\text{target},v}\}} > 0.5 \\ 0, & \text{otherwise} \end{cases} \tag{4.5}$$

The overall DEA is then computed as:

$$\text{DEA} = \frac{1}{N_{\text{videos}}} \sum_{v=1}^{N_{\text{videos}}} \text{dominant}_v \tag{4.6}$$

DEA With Neutral (DEA_{WN}) DEA_{WN} and AEA_{WN} extend DEA and AEA to the full 32-frame sequence by assigning a per-segment target: neutral in the onset (frames 0–9) and offset (frames 22–31), and the target emotion in the apex (frames 10–21). A video is considered correct under DEA_{WN} if more than 50% of all 32 frames satisfy their segment-specific target:

$$\text{dominant}_{v,\text{WN}} = \begin{cases} 1, & \text{if } \frac{1}{32}\left(\begin{array}{l} \sum_{t=0}^{9} \mathbb{1}_{\{\hat{e}_{v,t}=\text{neutral}\}} \\ +\sum_{t=10}^{21} \mathbb{1}_{\{\hat{e}_{v,t}=e_{\text{target},v}\}} \\ +\sum_{t=22}^{31} \mathbb{1}_{\{\hat{e}_{v,t}=\text{neutral}\}} \end{array}\right) > 0.5 \\ 0, & \text{otherwise} \end{cases} \tag{4.7}$$

Interpretation

DEA values range from 0 to 1, representing the fraction of videos that satisfy the majority-agreement criterion:

- **DEA**: Fraction of videos where the majority of central frames match the target emotion

- **DEA_{WN}**: Fraction of videos where the majority of all frames satisfy their segment-specific targets (neutral in onset/offset, target emotion in apex)
- **DEA = 1.0**: All videos satisfy majority agreement on the central frames
- **$\text{DEA}_{\text{WN}} = 1.0$**: All videos satisfy majority agreement across all frames with correct temporal structure
- **$\text{DEA}/\text{DEA}_{\text{WN}} = 0.0$**: No videos satisfy the majority-agreement criterion
- **DEA vs. AEA**: DEA is binary (video passes threshold or not), while AEA is continuous

A high DEA indicates that the model consistently matches the expected emotional behavior with strong majority agreement, rather than producing ambiguous or borderline cases.

4.4.4 Average Content Distance (ACD)

The Average Content Distance (ACD) metric assesses the model's ability to preserve visual content consistency throughout generated videos. It comprises two complementary sub-metrics: ACD-I (Identity Distance) and ACD-C (Content/Temporal Consistency Distance).

ACD-I: Identity Preservation

ACD-I measures how well the facial identity of the subject is preserved across all frames of the generated video.

Reference Frame The neutral face from frame 0 of the preprocessed video from the test set (after face detection, alignment, and resizing to 64×64) serves as the identity reference.

Face Recognition Models Three state-of-the-art face recognition models are employed to extract facial embeddings:

- **VGG-Face** [21]: 4096-dimensional embeddings based on a modified VGG-16 architecture

- **OpenFace** [3]: 128-dimensional embeddings based on FaceNet with Inception ResNet architecture
- **ArcFace** [8]: 512-dimensional embeddings using ResNet-100 with angular margin loss, representing state-of-the-art in face recognition

For each model, embeddings are extracted from both the reference frame and all generated frames using DeepFace:

$$\mathbf{emb}_{\text{ref}} = \text{DeepFace.represent}(\text{frame}_0^{\text{real}}, \text{model}) \tag{4.8}$$

$$\mathbf{emb}_{v,t} = \text{DeepFace.represent}(\text{frame}_t^{\text{gen},v}, \text{model}), \quad t = 0, \ldots, 31 \tag{4.9}$$

Distance Metrics Two distance measures are computed between embeddings:

- **Euclidean distance**:

$$d_{\text{eucl}}(\mathbf{emb}_{\text{ref}}, \mathbf{emb}_{v,t}) = \|\mathbf{emb}_{\text{ref}} - \mathbf{emb}_{v,t}\|_2 \tag{4.10}$$

- **Cosine distance**:

$$d_{\cos}(\mathbf{emb}_{\text{ref}}, \mathbf{emb}_{v,t}) = 1 - \frac{\mathbf{emb}_{\text{ref}} \cdot \mathbf{emb}_{v,t}}{\|\mathbf{emb}_{\text{ref}}\|_2 \|\mathbf{emb}_{v,t}\|_2} \tag{4.11}$$

Recognition Rate For each model and distance metric, model-specific thresholds are used to determine whether a face is recognized as the same identity:

- **VGG-Face**: cosine ≤ 0.4, euclidean ≤ 0.6
- **OpenFace**: cosine ≤ 0.1, euclidean ≤ 0.55
- **ArcFace**: cosine ≤ 0.1, euclidean ≤ 4.15

The recognition rate is computed as:

$$\text{Recognition Rate} = \frac{1}{N_{\text{videos}} \times 32} \sum_{v=1}^{N_{\text{videos}}} \sum_{t=0}^{31} \mathbb{1}_{\{d(\mathbf{emb}_{\text{ref},v}, \mathbf{emb}_{v,t}) \leq \tau\}} \tag{4.12}$$

where τ is the appropriate threshold for the chosen model and distance metric.

Average Distance The ACD-I is computed as the mean distance across all videos and all frames:

$$\text{ACD-I} = \frac{1}{N_{\text{videos}} \times 32} \sum_{v=1}^{N_{\text{videos}}} \sum_{t=0}^{31} d(\mathbf{emb}_{\text{ref},v}, \mathbf{emb}_{v,t}) \tag{4.13}$$

ACD-C: Temporal Consistency

ACD-C evaluates the smoothness and consistency of transitions between consecutive frames in the generated videos.

Consecutive Frame Comparison For each video, distances are computed between all pairs of consecutive frames:

$$d_{v,t} = d(\mathbf{emb}_{v,t}, \mathbf{emb}_{v,(t+1) \pmod{32}}), \quad t = 0, \ldots, 31 \tag{4.14}$$

where the last frame wraps around to the first one, creating a cyclic sequence (32 pairs total per video).

Average Temporal Distance The ACD-C is computed as:

$$\text{ACD-C} = \frac{1}{N_{\text{videos}} \times 32} \sum_{v=1}^{N_{\text{videos}}} \sum_{t=0}^{31} d(\mathbf{emb}_{v,t}, \mathbf{emb}_{v,(t+1) \pmod{32}}) \tag{4.15}$$

Interpretation

The ACD metrics provide complementary information about content preservation:

- **ACD-I Distance**: Lower values indicate better identity preservation; range varies by model
- **ACD-I Recognition Rate**: Range [0, 1]; higher values (closer to 1.0) indicate better identity preservation
- **ACD-C Distance**: Lower values indicate smoother, more consistent temporal transitions

Together, ACD-I and ACD-C ensure that generated videos maintain both the subject's facial identity and temporal coherence throughout the sequence.

Chapter 5

Results

5.1 Introduction

This chapter reports the quantitative and qualitative evaluation of all proposed configurations on the MUG test set (158 videos from 10 unseen identities). The goal is to understand how the two main modifications—(i) replacing the GAN image generator with a diffusion model, and (ii) shifting the emotion conditioning from the frame generator to the motion-code generator—affect (a) overall spatio-temporal quality, (b) emotion expressiveness, and (c) identity preservation.

For ease of comprehension and to simplify the reading of the results, Table 5.1 reports again the canonical names of the compared settings introduced in Chapter 3.

`Canonical name`	**Description**
`baseline`	Original VICEGAN with GAN-based frame synthesis.
`onlyEM`	Conditioning shift applied to the motion generator, while keeping GAN-based frame synthesis.
`onlyDM`	Diffusion model replacing GAN.
`EM-DM`	Diffusion-based frame synthesis combined with the conditioning shift, with emotion injected in the motion generator.
`both_retrievedMcs`	Diffusion model driven by motion codes *retrieved* from real training sequences using an embedding network.
`LFDM`	State-of-the-art reference method.

Table 5.1: Canonical names and short descriptions of the compared model configurations.

5.1.1 Fréchet Video Distance (FVD)

Table 5.2 reports FVD; for the formal definition and computation details, see Section 4.4.1 in Chapter 4. Here, it is sufficient to recall that lower values indicate generated videos that are closer to the real-data distribution, and therefore better overall perceptual quality at the distribution level. Overall, replacing GAN-based synthesis with a diffusion model leads to a strong improvement in video quality.

The best overall score is obtained by **onlyDM** (FVD = 100), followed by **EM-DM** (FVD = 120). In contrast, the GAN-based **baseline** has a substantially higher FVD (= 430). The conditioning shift alone also improves quality in the GAN setting: **onlyEM** reduces FVD from 430 to 346. The retrieval-driven configuration **both_retrievedMcs** achieves an intermediate FVD (= 287), close to **LFDM** (= 291).

	All↓	**Angry↓**	**Disgust↓**	**Surprise↓**	**Sad↓**	**Happy↓**	**Fear↓**
LFDM	291	395	387	431	446	467	395
baseline	430	580	494	566	683	491	646
onlyEM	346	465	461	520	451	397	555
onlyDM	**100**	**186**	**227**	**245**	203	**206**	**230**
EM-DM	120	203	237	247	**194**	221	277
both_retrievedMcs	287	650	403	402	296	335	678

Table 5.2: Fréchet Video Distance (FVD, lower is better) results obtained by different models on the test set.

Overall, these results support the conclusion that diffusion-based synthesis produces higher-quality videos in terms of distributional realism (as measured by FVD) and visual fidelity (as observed in qualitative samples). Temporal coherence is analyzed separately through ACD-C and the temporal AEA/ACD-I trends.

5.1.2 Emotion Accuracy Metrics

Table 5.3 reports DEA, AEA, and their WN variants; the formal definitions are given in Sections 4.4.2 and 4.4.3 of Chapter 4. In this chapter, these metrics are interpreted as measures of how well the generated sequence expresses the intended emotion: higher values indicate better emotion control, while the WN variants additionally reward the correct neutral→emotion→neutral temporal structure over the full sequence.

		DEA↑	AEA↑	DEA WN↑	AEA WN↑
all	LFDM	0.58	0.60	0.23	0.37
	baseline	0.74	0.70	**0.63**	**0.58**
	onlyEM	**0.76**	**0.76**	0.54	0.52
	onlyDM	0.66	0.64	0.46	0.49
	EM-DM	0.63	0.58	0.56	0.52
	both_retrievedMcs	0.75	0.74	0.53	0.52
angry	LFDM	0.43	0.47	0.17	0.32
	baseline	0.73	0.72	**0.67**	**0.57**
	onlyEM	0.50	0.57	0.40	0.43
	onlyDM	0.60	0.60	0.40	0.47
	EM-DM	0.50	0.51	0.50	0.50
	both_retrievedMcs	**0.83**	**0.81**	0.37	0.44
disgust	LFDM	0.63	0.63	0.17	0.34
	baseline	0.83	0.72	0.50	**0.53**
	onlyEM	**0.93**	**0.91**	**0.53**	0.52
	onlyDM	0.73	0.69	0.37	0.43
	EM-DM	0.77	0.66	0.43	0.46
	both_retrievedMcs	0.43	0.51	0.23	0.39
surprise	LFDM	0.59	0.60	0.21	0.36
	baseline	0.66	0.66	0.59	0.57
	onlyEM	0.90	0.82	0.55	0.56
	onlyDM	0.59	0.57	0.41	0.48
	EM-DM	0.55	0.53	0.62	0.54
	both_retrievedMcs	**0.97**	**0.91**	**0.83**	**0.68**
sad	LFDM	0.43	0.46	0.19	0.33
	baseline	0.52	0.52	**0.62**	**0.53**
	onlyEM	**0.62**	**0.64**	0.52	0.49
	onlyDM	0.52	0.52	0.52	0.44
	EM-DM	0.43	0.42	0.57	0.43
	both_retrievedMcs	0.48	0.53	0.57	0.44
happy	LFDM	0.90	0.87	0.39	0.51
	baseline	**1.00**	0.97	**0.87**	**0.72**
	onlyEM	**1.00**	**1.00**	0.74	0.62
	onlyDM	0.94	0.90	0.77	0.64
	EM-DM	0.97	0.87	0.81	0.69
	both_retrievedMcs	**1.00**	0.98	0.81	0.66
fear	LFDM	0.35	0.42	0.24	0.36
	baseline	0.53	0.45	**0.41**	**0.49**
	onlyEM	0.41	0.47	**0.41**	0.46
	onlyDM	0.47	0.42	0.18	0.44
	EM-DM	0.35	0.29	0.29	0.43
	both_retrievedMcs	**0.65**	**0.55**	0.24	**0.49**

Table 5.3: Quality of emotion generation (R2) results obtained by different models on the test set. Comparison of Dominant Emotion Accuracy (DEA), Average Emotion Accuracy (AEA), and their With Neutral variants (WN), which incorporate the neutral–apex–neutral structure of the full sequence.

The main trend is that GAN-based configurations provide higher emotion recognition scores than diffusion-based ones. In particular, **onlyEM** achieves the best overall performance (DEA = 0.76, AEA = 0.76), closely followed by **both_retrievedMcs** (DEA = 0.75, AEA = 0.74) and the **baseline** (DEA = 0.74, AEA = 0.70).

This behavior is expected for adversarial training: the GAN pipeline includes a discriminator that explicitly encourages frames to look plausible and, in practice, to be recognized correctly by an external emotion classifier. However, the compar-

ison with FVD suggests a critical caveat: higher emotion-classification accuracy does not necessarily imply better perceptual quality. In some cases, distortions, exaggerated facial deformations, or high-frequency artifacts can increase the confidence of an automated classifier while reducing realism. For this reason, emotion metrics should be interpreted jointly with quality and consistency metrics, and complemented with qualitative inspection.

5.1.3 Identity Preservation (ACD-I) and Average Emotion Accuracy (AEA) Trends

To better understand *when* models express emotions and *how* identity changes over time, we analyze the temporal profiles of AEA and ACD-I across the 32 generated frames. For the formal definitions of these quantities, see Sections 4.4.2 and 4.4.4 in Chapter 4. Here, the interpretation is straightforward: an effective model should show an AEA peak around the apex frames (10–21), while ACD-I should remain as stable as possible over time, since strong oscillations typically correspond to identity drift during expression generation.

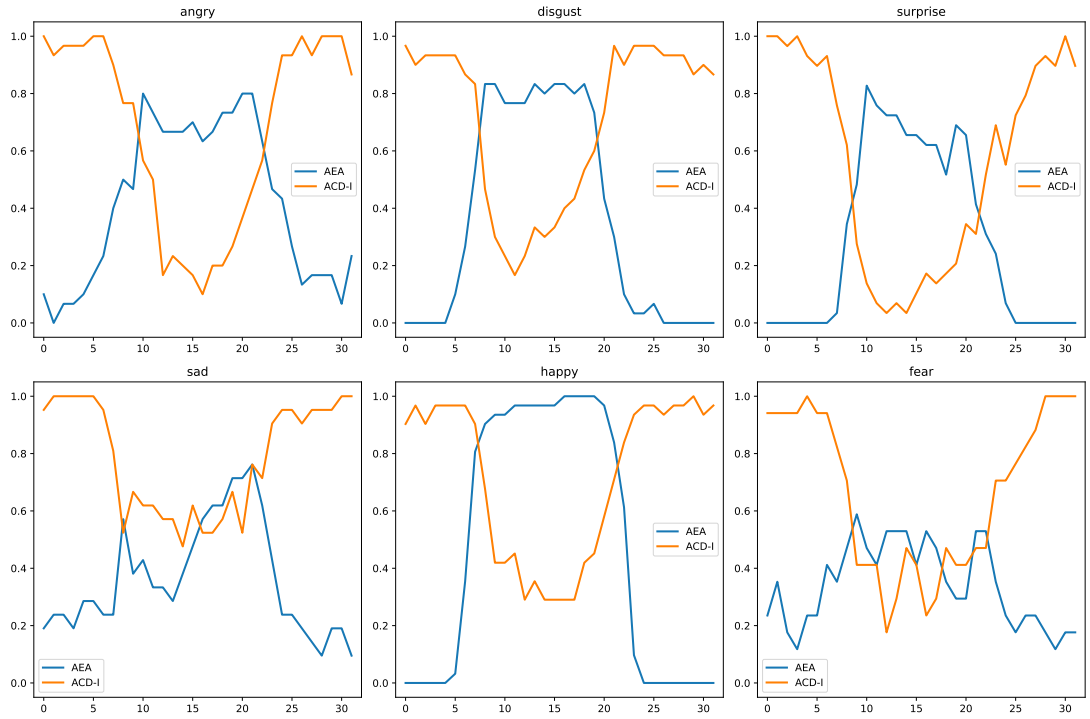


Figure 5.1: Temporal analysis of average emotion accuracy (AEA) and identity consistency (ACD-I) for the VICEGAN baseline model.

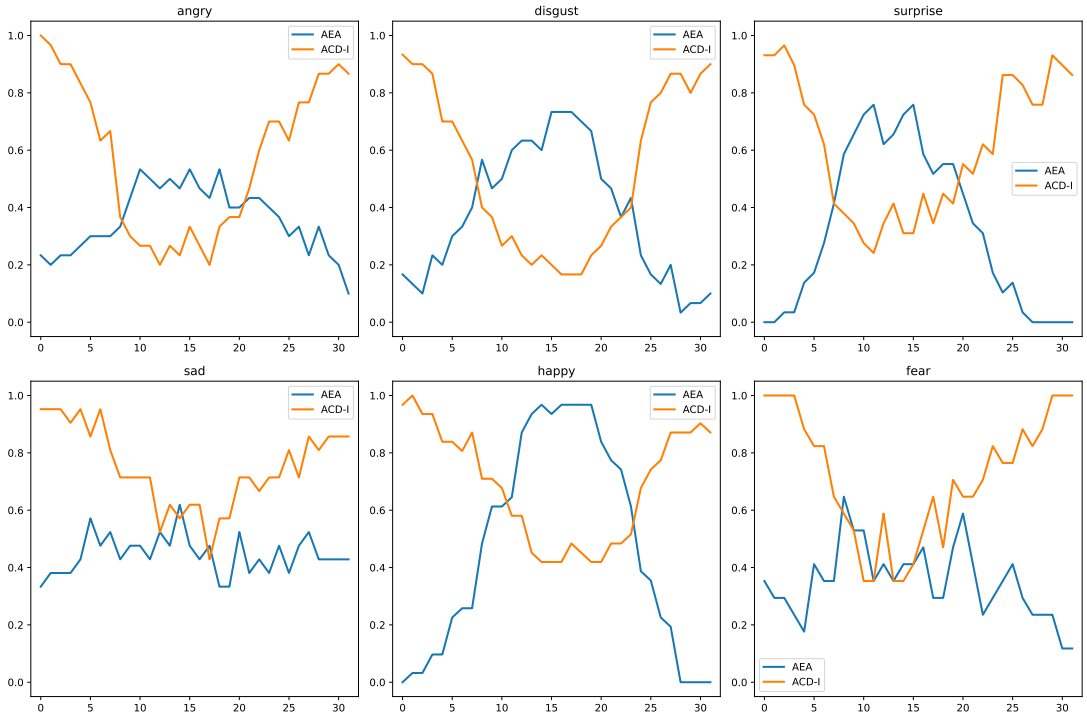


Figure 5.2: Temporal analysis of average emotion accuracy (AEA) and identity consistency (ACD-I) for the LFDM model.

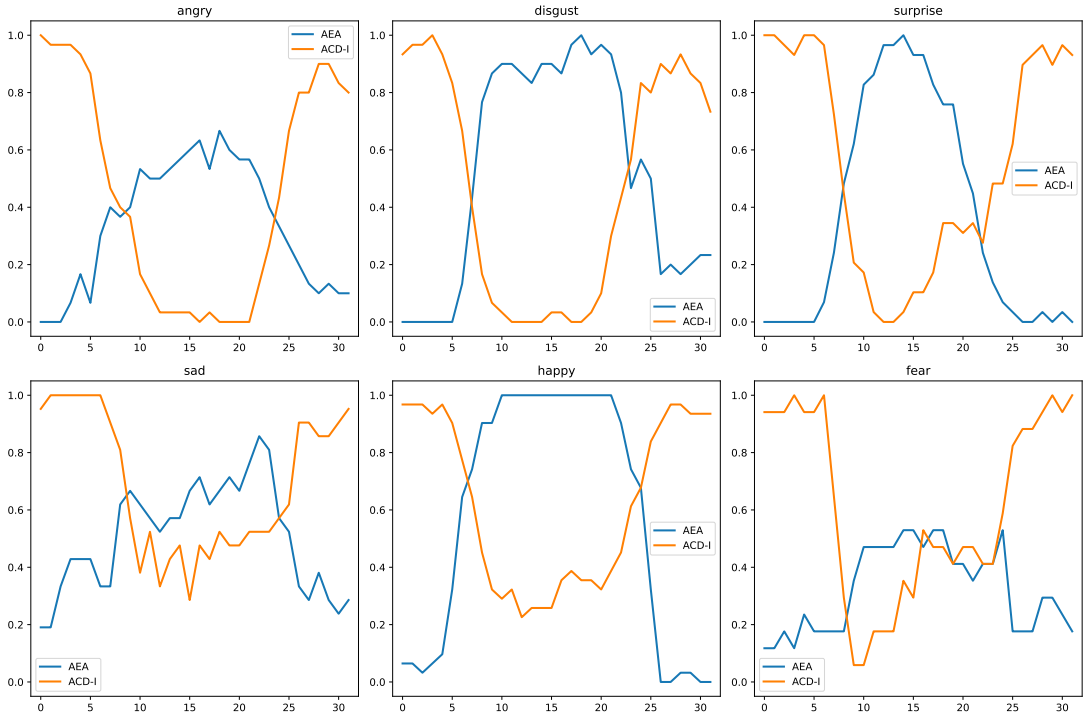


Figure 5.3: Temporal analysis of average emotion accuracy (AEA) and identity consistency (ACD-I) for the onlyEM model.

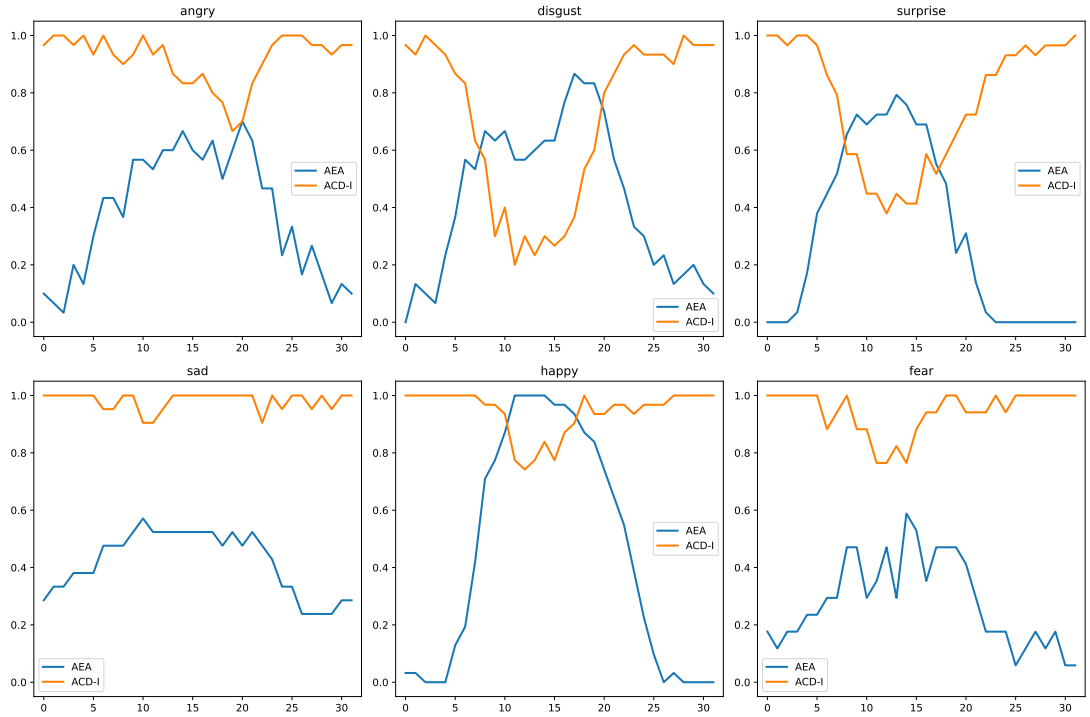


Figure 5.4: Temporal analysis of average emotion accuracy (AEA) and identity consistency (ACD-I) for the onlyDM model.

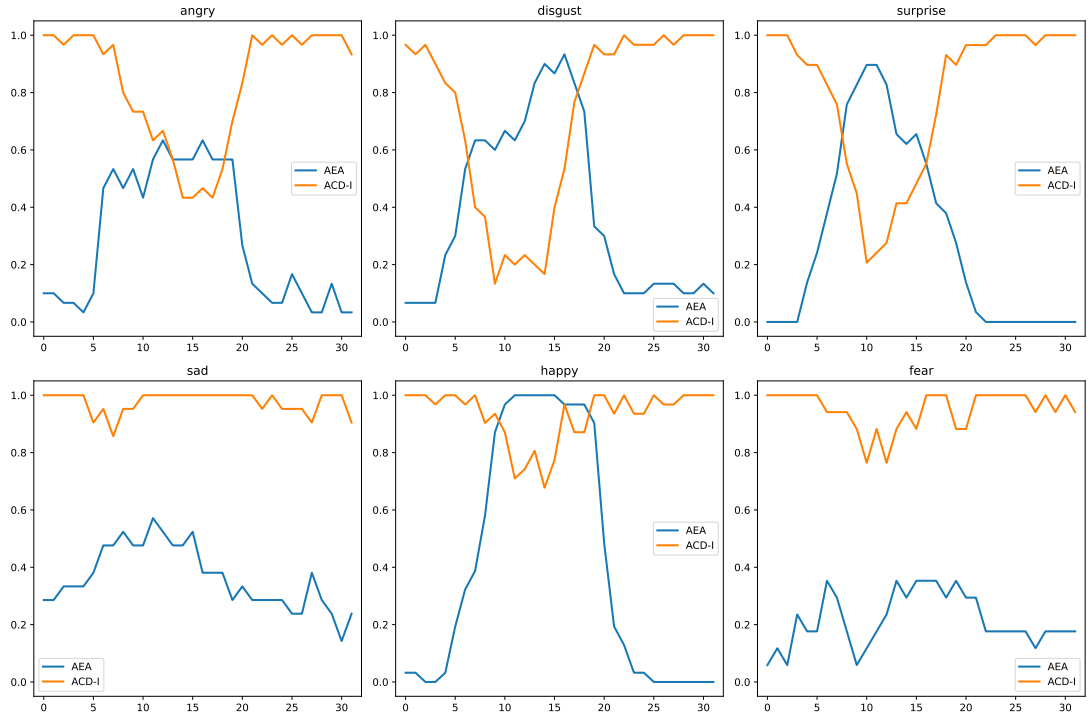


Figure 5.5: Temporal analysis of average emotion accuracy (AEA) and identity consistency (ACD-I) for the diffusion-based model retrained with the emotion-conditioned motion generator (EM-DM).

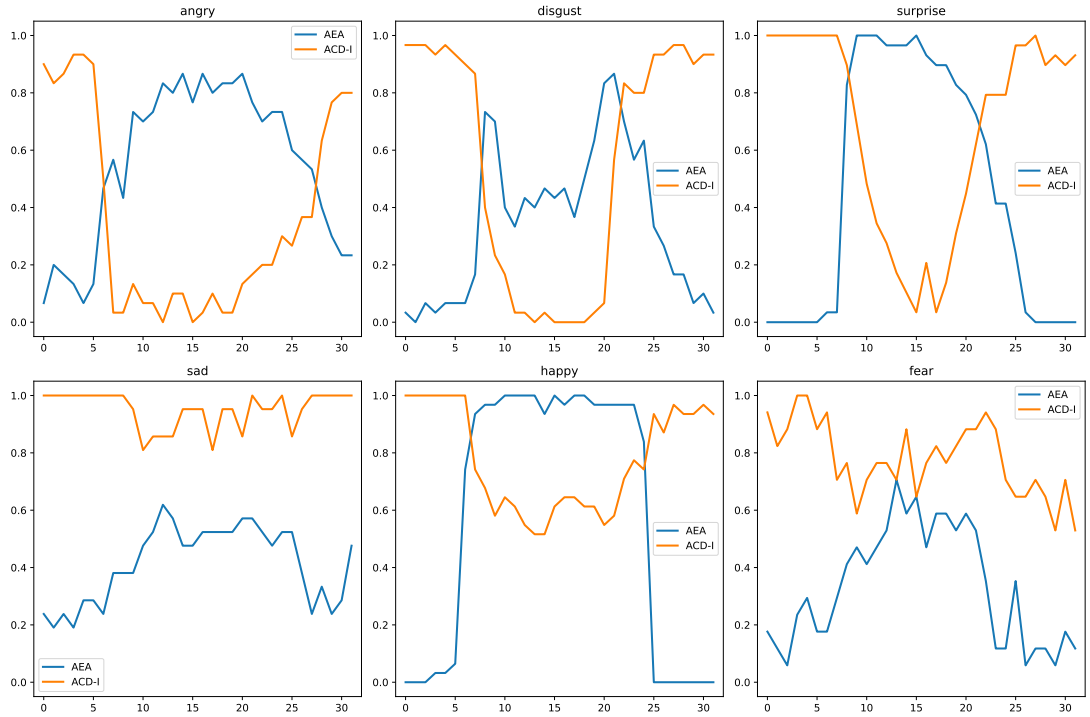


Figure 5.6: Temporal analysis of average emotion accuracy (AEA) and identity consistency (ACD-I) for the diffusion model driven by retrieved motion codes (both_retrievedMcs).

Across models, the plots highlight a clear trade-off: GAN-based variants tend to express emotions more strongly, as indicated by higher AEA peaks, while diffusion-based variants preserve identity more reliably, as reflected in more stable ACD-I trends. In this comparison, **LFDM** generally lies between the GAN variants and the diffusion-based models, showing intermediate behavior across both dimensions. In contrast, **both_retrievedMcs** does not exhibit the desired emotional peak and shows higher variability across generated frames, suggesting less stable emotional dynamics.

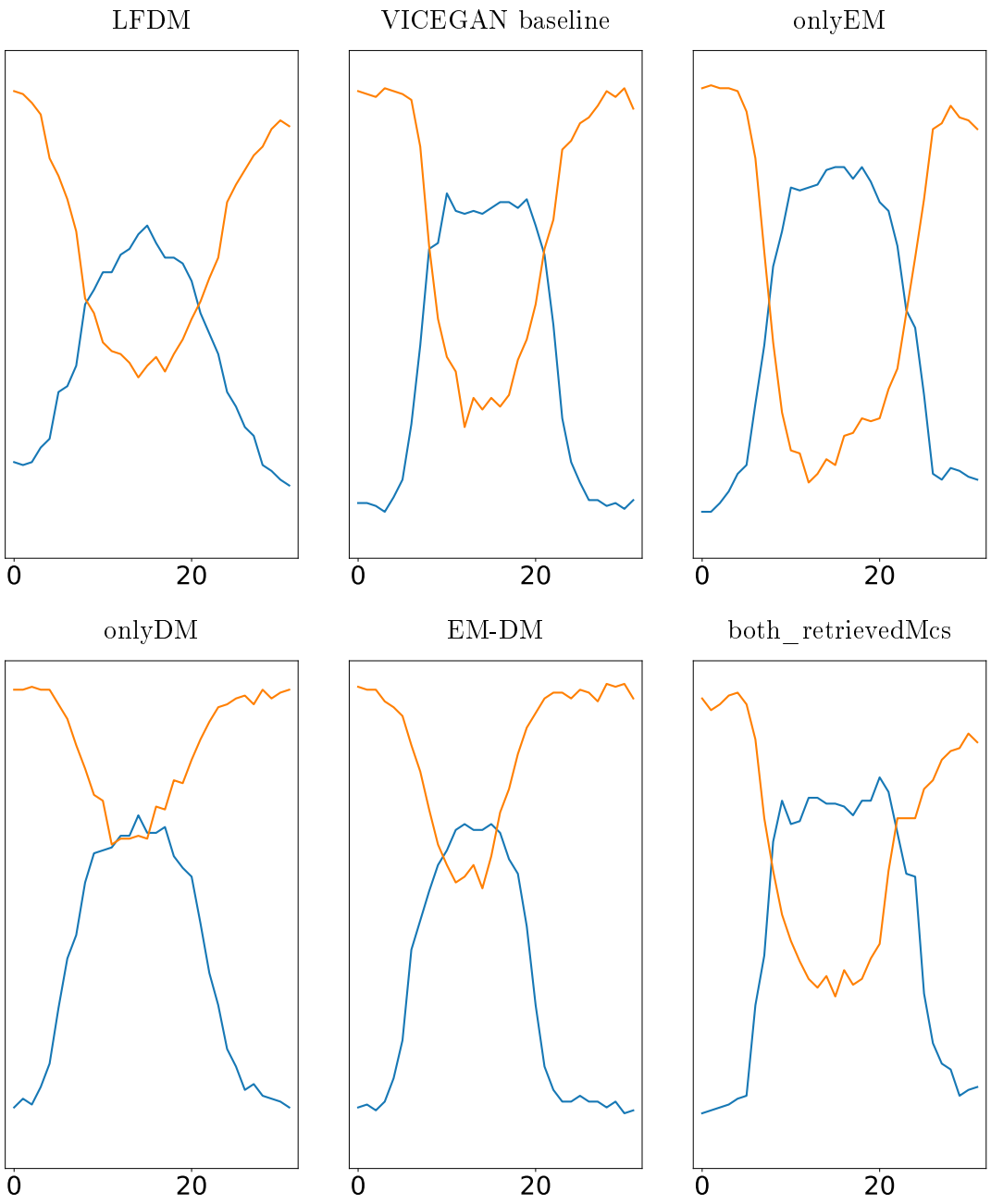


Figure 5.7: Overall AEA and ACD-I trends (aggregated over all emotions) for all compared models and the original dataset. Each panel shows the temporal profile of emotion accuracy and identity consistency across the 32 frames, enabling a direct visual comparison of expression strength and identity preservation.

5.1.4 Identity Consistency

Table 5.4 reports ACD-I; for the formal definition, see Section 4.4.4 in Chapter 4. In the discussion below, ACD-I is interpreted as a measure of identity preservation

under expression changes: lower distances and higher recognition rates indicate that the generated face remains closer to the source identity.

Diffusion-based models achieve the most consistent identity metrics. In the aggregated setting, **onlyDM** (and, very closely, **EM-DM**) shows the lowest distances and the highest recognition rates under both OpenFace and ArcFace (Table 5.4), indicating stronger identity preservation under expression changes than other models. **both_retrievedMcs** outperforms both the VICEGAN **baseline** and **LFDM** on several identity metrics, but it is still consistently outperformed by the other diffusion-based variants in this comparison. Overall, these results support the conclusion that diffusion-based synthesis preserves identity more reliably under expression changes.

		OpenFace				**ArcFace**				**VGG-Face**			
		Cos_d↓	Cos_rr↑	Euc_d↓	Euc_rr↑	Cos_d↓	Cos_rr↑	Euc_d↓	Euc_rr↑	Cos_d↓	Cos_rr↑	Euc_d↓	Euc_rr↑
all	LFDM	0.14	0.40	0.51	0.64	0.29	0.00	3.54	0.85	0.16	1.00	0.31	**1.00**
	baseline	0.13	0.44	0.49	0.68	0.45	0.00	4.40	0.37	0.20	0.98	0.35	1.00
	onlyEM	0.15	0.40	0.52	0.60	0.46	0.00	4.47	0.33	0.22	0.96	0.36	1.00
	onlyDM	0.08	0.72	0.36	**0.87**	**0.16**	0.38	**2.58**	**0.94**	0.08	**1.00**	0.22	1.00
	EM-DM	**0.08**	**0.73**	**0.35**	0.86	0.16	**0.39**	2.60	0.94	**0.08**	**1.00**	**0.21**	1.00
	both_retrievedMcs	0.12	0.52	0.45	0.68	0.24	0.20	3.18	0.81	0.13	0.99	0.27	1.00
angry	LFDM	0.15	0.33	0.53	0.58	0.29	0.00	3.55	0.89	0.15	**1.00**	0.30	**1.00**
	baseline	0.14	0.47	0.50	0.68	0.44	0.00	4.39	0.40	0.18	0.99	0.33	**1.00**
	onlyEM	0.17	0.34	0.56	0.49	0.45	0.00	4.42	0.31	0.20	**1.00**	0.34	**1.00**
	onlyDM	**0.08**	**0.69**	**0.36**	**0.89**	**0.17**	0.31	**2.66**	**0.96**	**0.08**	**1.00**	0.21	**1.00**
	EM-DM	0.08	0.68	0.37	0.84	0.18	**0.35**	2.71	0.93	0.08	**1.00**	**0.21**	**1.00**
	both_retrievedMcs	0.18	0.28	0.56	0.40	0.38	0.13	3.95	0.51	0.18	0.97	0.31	**1.00**
disgust	LFDM	0.16	0.38	0.54	0.53	0.31	0.00	3.73	0.70	0.18	1.00	0.32	**1.00**
	baseline	0.13	0.49	0.48	0.71	0.43	0.00	4.34	0.38	0.20	0.98	0.35	0.99
	onlyEM	0.17	0.36	0.56	0.50	0.47	0.00	4.54	0.30	0.25	0.91	0.39	1.00
	onlyDM	0.10	0.60	0.40	0.74	0.19	0.33	2.88	**0.87**	0.12	**1.00**	0.26	0.99
	EM-DM	**0.09**	**0.60**	**0.39**	**0.75**	**0.18**	**0.39**	**2.79**	0.87	**0.11**	**1.00**	**0.25**	0.99
	both_retrievedMcs	0.13	0.47	0.47	0.60	0.25	0.22	3.25	0.71	0.14	0.99	0.28	0.99
surprise	LFDM	0.16	0.37	0.53	0.61	0.30	0.00	3.61	0.81	0.15	**1.00**	0.31	**1.00**
	baseline	0.16	0.32	0.55	0.56	0.50	0.00	4.64	0.26	0.23	0.96	0.38	1.00
	onlyEM	0.16	0.37	0.54	0.58	0.50	0.00	4.64	0.27	0.21	0.98	0.37	**1.00**
	onlyDM	0.11	0.57	0.43	0.78	0.20	0.29	2.87	0.87	0.10	**1.00**	0.25	**1.00**
	EM-DM	**0.10**	**0.66**	**0.40**	**0.79**	**0.20**	**0.34**	**2.82**	**0.87**	**0.09**	**1.00**	**0.23**	**1.00**
	both_retrievedMcs	0.13	0.49	0.46	0.68	0.22	0.19	3.03	0.86	0.10	**1.00**	0.26	**1.00**
sad	LFDM	0.12	0.47	0.47	0.75	0.26	0.00	3.36	0.95	0.16	**1.00**	0.30	**1.00**
	baseline	0.11	0.53	0.45	0.79	0.43	0.00	4.23	0.48	0.18	1.00	0.33	**1.00**
	onlyEM	0.12	0.51	0.46	0.74	0.44	0.00	4.29	0.44	0.20	**1.00**	0.34	**1.00**
	onlyDM	**0.04**	**0.94**	**0.26**	**0.97**	**0.10**	**0.63**	**2.06**	**1.00**	0.05	**1.00**	0.17	**1.00**
	EM-DM	0.04	0.93	0.26	0.96	0.10	0.57	2.14	**1.00**	**0.05**	**1.00**	**0.17**	**1.00**
	both_retrievedMcs	0.06	0.89	0.31	0.94	0.15	0.37	2.56	0.98	0.08	**1.00**	0.21	**1.00**
happy	LFDM	0.13	0.43	0.48	0.69	0.26	0.00	3.44	0.91	0.16	**1.00**	0.31	**1.00**
	baseline	0.12	0.45	0.47	0.72	0.43	0.00	4.32	0.40	0.22	0.95	0.36	**1.00**
	onlyEM	0.13	0.46	0.48	0.70	0.43	0.00	4.38	0.37	0.24	0.92	0.38	1.00
	onlyDM	**0.06**	**0.83**	**0.32**	**0.98**	**0.12**	**0.42**	**2.37**	**1.00**	**0.08**	**1.00**	**0.21**	**1.00**
	EM-DM	0.06	0.79	0.33	0.94	0.13	0.40	2.44	0.99	0.08	**1.00**	0.21	**1.00**
	both_retrievedMcs	0.09	0.63	0.41	0.82	0.18	0.20	2.82	0.99	0.13	**1.00**	0.27	**1.00**
fear	LFDM	0.13	0.48	0.48	0.72	0.28	0.00	3.47	0.90	0.13	**1.00**	0.30	**1.00**
	baseline	0.14	0.38	0.50	0.67	0.48	0.00	4.50	0.32	0.20	1.00	0.36	0.99
	onlyEM	0.13	0.41	0.49	0.64	0.49	0.00	4.55	0.31	0.20	**1.00**	0.35	**1.00**
	onlyDM	0.07	0.77	**0.34**	0.93	**0.14**	**0.41**	**2.44**	**0.98**	**0.06**	**1.00**	**0.20**	**1.00**
	EM-DM	**0.07**	**0.79**	0.35	**0.94**	0.15	0.36	2.52	0.98	0.07	**1.00**	0.20	**1.00**
	both_retrievedMcs	0.12	0.44	0.47	0.77	0.26	0.05	3.35	0.89	0.12	**1.00**	0.28	1.00

Table 5.4: Identity consistency (R4) results obtained by different models on the test set. Comparison of Cosine Distance (dist., lower is better) and Recognition Rate (r. rate, higher is better) across OpenFace, ArcFace, and VGG-Face embeddings.

5.1.5 Frame Continuity and Temporal Coherence

Table 5.5 reports ACD-C; the formal definition is given in Section 4.4.4 of Chapter 4. Here, we use it only as a compact indicator of short-term temporal smoothness: lower distance and higher recognition rates correspond to fewer discontinuities between consecutive frames.

		OpenFace				**ArcFace**				**VGG-Face**			
		Cos_d↓	Cos_rr↑	Euc_d↓	Euc_rr↑	Cos_d↓	Cos_rr↑	Euc_d↓	Euc_rr↑	Cos_d↓	Cos_rr↑	Euc_d↓	Euc_rr↑
all	LFDM	0.04	0.92	0.25	0.97	0.10	0.63	1.89	1.00	0.04	**1.00**	0.14	**1.00**
	baseline	0.04	0.94	0.24	0.98	0.11	0.63	1.88	0.99	0.04	**1.00**	0.16	**1.00**
	onlyEM	0.04	0.92	0.25	0.97	0.10	0.68	1.82	0.98	0.04	**1.00**	0.15	**1.00**
	onlyDM	**0.03**	**0.94**	**0.23**	**0.98**	**0.08**	**0.76**	**1.76**	**1.00**	**0.03**	**1.00**	**0.14**	**1.00**
	EM-DM	0.04	0.93	0.24	0.97	0.08	0.72	1.84	1.00	0.04	**1.00**	0.14	**1.00**
	both_retrievedMcs	0.05	0.86	0.30	0.95	0.13	0.48	2.25	0.99	0.05	**1.00**	0.18	**1.00**
angry	LFDM	0.03	0.94	0.24	0.99	0.10	0.66	1.83	1.00	0.04	**1.00**	0.14	**1.00**
	baseline	0.04	0.94	0.24	0.99	0.10	0.67	1.81	0.99	0.04	**1.00**	0.15	**1.00**
	onlyEM	0.04	0.94	0.25	0.98	0.09	0.75	**1.68**	0.98	0.04	**1.00**	0.14	**1.00**
	onlyDM	**0.03**	**0.96**	**0.23**	**0.99**	**0.07**	**0.79**	1.71	**1.00**	**0.03**	**1.00**	**0.13**	**1.00**
	EM-DM	0.04	0.93	0.25	0.97	0.08	0.74	1.81	1.00	0.03	**1.00**	0.14	**1.00**
	both_retrievedMcs	0.07	0.78	0.35	0.91	0.16	0.29	2.51	0.96	0.07	**1.00**	0.20	**1.00**
disgust	LFDM	0.03	0.93	0.23	0.98	0.09	0.68	1.88	0.99	0.04	**1.00**	**0.14**	**1.00**
	baseline	0.03	**0.96**	0.22	**0.99**	0.09	0.71	1.79	0.99	0.04	**1.00**	0.15	**1.00**
	onlyEM	0.04	0.93	0.25	0.97	0.09	0.72	1.80	0.98	0.04	**1.00**	0.15	**1.00**
	onlyDM	**0.03**	0.95	**0.22**	0.99	**0.07**	**0.80**	**1.77**	**1.00**	**0.03**	**1.00**	0.14	**1.00**
	EM-DM	0.04	0.93	0.24	0.97	0.08	0.77	1.83	**1.00**	0.04	**1.00**	0.15	**1.00**
	both_retrievedMcs	0.04	0.94	0.25	0.98	0.11	0.56	2.18	0.99	0.05	**1.00**	0.17	**1.00**
surprise	LFDM	0.04	0.90	**0.26**	0.96	0.11	0.57	1.96	1.00	0.04	**1.00**	**0.15**	**1.00**
	baseline	0.04	0.91	0.27	0.97	0.13	0.54	2.07	0.98	0.05	**1.00**	0.17	**1.00**
	onlyEM	**0.04**	0.89	0.26	**0.97**	0.11	0.61	1.93	0.98	0.04	**1.00**	0.16	**1.00**
	onlyDM	0.04	**0.91**	0.26	0.96	**0.10**	**0.65**	**1.89**	1.00	**0.04**	**1.00**	0.16	**1.00**
	EM-DM	0.05	0.89	0.27	0.95	0.11	0.57	2.02	0.99	0.04	**1.00**	0.16	**1.00**
	both_retrievedMcs	0.06	0.86	0.31	0.95	0.13	0.40	2.25	**1.00**	0.05	**1.00**	0.18	**1.00**
sad	LFDM	0.04	0.93	0.24	0.98	0.09	0.70	1.78	1.00	0.04	**1.00**	0.13	**1.00**
	baseline	0.03	0.96	0.24	0.99	0.10	0.63	1.84	0.99	0.04	**1.00**	0.15	**1.00**
	onlyEM	0.04	0.93	0.25	0.97	0.10	0.65	1.79	0.98	0.04	**1.00**	0.14	**1.00**
	onlyDM	0.03	0.94	0.21	0.99	**0.06**	**0.81**	**1.59**	**1.00**	**0.03**	**1.00**	**0.12**	**1.00**
	EM-DM	**0.03**	0.96	**0.20**	0.99	0.06	**0.81**	1.65	**1.00**	0.03	**1.00**	0.12	**1.00**
	both_retrievedMcs	0.03	**0.96**	0.23	**0.99**	0.08	0.75	1.86	**1.00**	0.03	**1.00**	0.14	**1.00**
happy	LFDM	0.04	0.91	0.25	0.97	0.10	0.63	1.90	**1.00**	0.04	**1.00**	0.15	**1.00**
	baseline	0.04	0.92	0.24	0.97	0.10	0.65	1.83	0.99	0.04	**1.00**	0.15	**1.00**
	onlyEM	0.04	0.92	0.24	0.97	0.09	0.74	1.75	0.99	0.04	**1.00**	0.15	**1.00**
	onlyDM	**0.03**	**0.95**	**0.22**	**0.98**	**0.07**	**0.81**	**1.70**	**1.00**	**0.03**	**1.00**	**0.13**	**1.00**
	EM-DM	0.03	0.93	0.23	0.97	0.07	0.80	1.77	**1.00**	0.03	**1.00**	0.14	**1.00**
	both_retrievedMcs	0.05	0.88	0.28	0.94	0.10	0.64	2.04	1.00	0.04	**1.00**	0.16	**1.00**
fear	LFDM	0.05	0.86	0.28	0.94	0.12	0.51	2.03	**1.00**	0.04	**1.00**	0.16	**1.00**
	baseline	0.04	0.92	0.26	0.97	0.12	0.56	2.00	0.99	0.05	**1.00**	0.17	**1.00**
	onlyEM	0.05	0.89	0.28	0.97	0.13	0.52	2.05	0.99	0.05	**1.00**	0.17	**1.00**
	onlyDM	**0.04**	**0.95**	**0.25**	**0.98**	**0.09**	**0.64**	**1.91**	1.00	**0.04**	**1.00**	**0.15**	**1.00**
	EM-DM	0.04	0.93	0.25	0.98	0.10	0.63	1.96	**1.00**	0.04	**1.00**	0.16	**1.00**
	both_retrievedMcs	0.08	0.75	0.37	0.90	0.20	0.14	2.80	0.98	0.08	**1.00**	0.22	**1.00**

Table 5.5: Frame continuity consistency (ACD-C) results obtained by different models on the test set. Comparison of Cosine Distance (dist., lower is better) and Recognition Rate (r. rate, higher is better) across OpenFace, ArcFace, and VGG-Face embeddings, measured between consecutive frames.

All configurations achieve high temporal continuity, with recognition rates close to 1.00 in most cases. The **onlyDM** model achieves the best overall OpenFace and ArcFace distances. **LFDM**, **onlyDM**, and **EM-DM** are close to each other and are generally slightly better than the VICEGAN-based configurations. In contrast, only **both_retrievedMcs** shows a clear drop in temporal continuity, with higher distances and lower recognition rates across all embeddings; this suggests that the retrieved motion codes, in this specific training setting, may

not be fully compatible with the diffusion-based frame generator, leading to less stable transitions between frames.

5.1.6 Qualitative Results: Generated Samples

This section presents visual examples of generated video sequences to complement the quantitative evaluation. For each model configuration, Figure 5.9 through Figure 5.13 show representative frames sampled uniformly from a generated 32-frame video sequence. Each figure displays 10 frames (64×64 pixels) arranged horizontally, illustrating the temporal evolution from neutral expression to the target emotion and back to neutral.

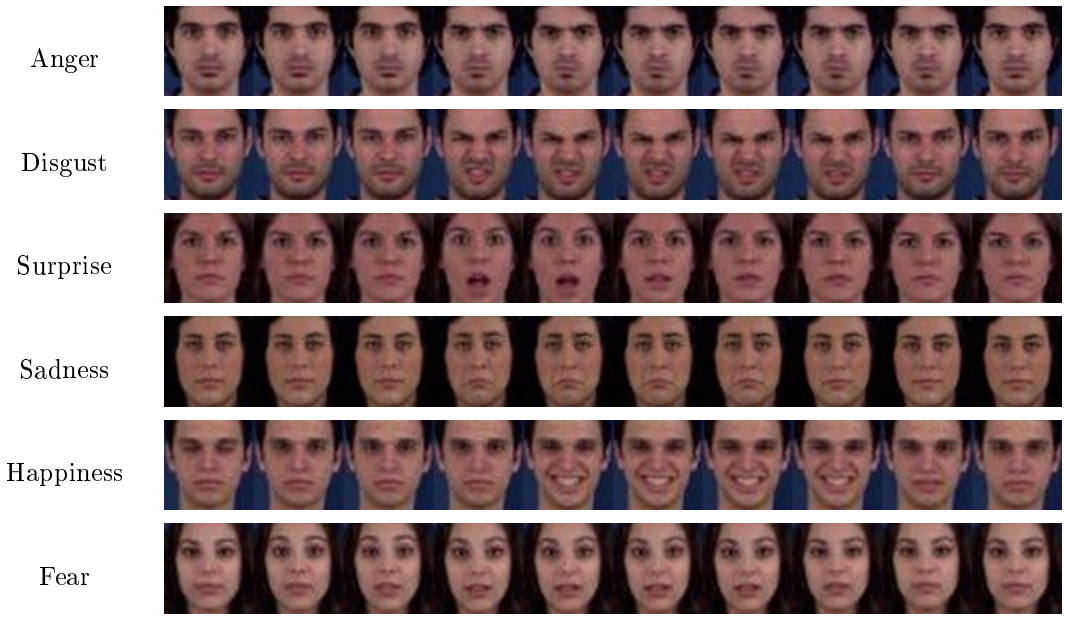


Figure 5.8: Generated video sequences from the LFDM model. Each row corresponds to a different emotion class, showing 10 uniformly sampled frames.

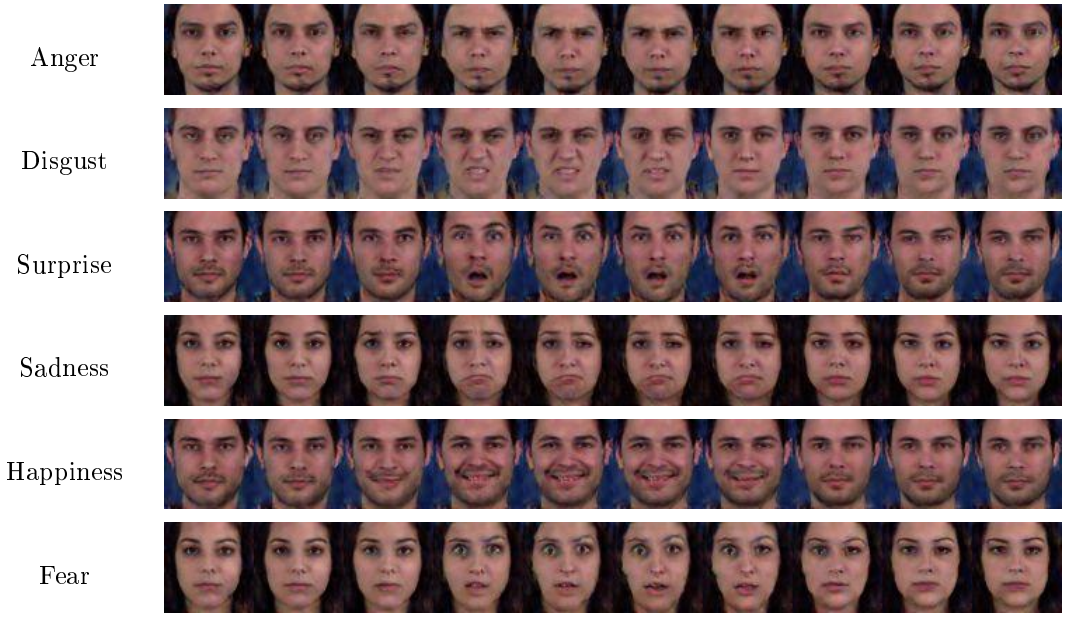


Figure 5.9: Generated video sequences from the VICEGAN baseline model. Each row corresponds to a different emotion class, showing 10 uniformly sampled frames.

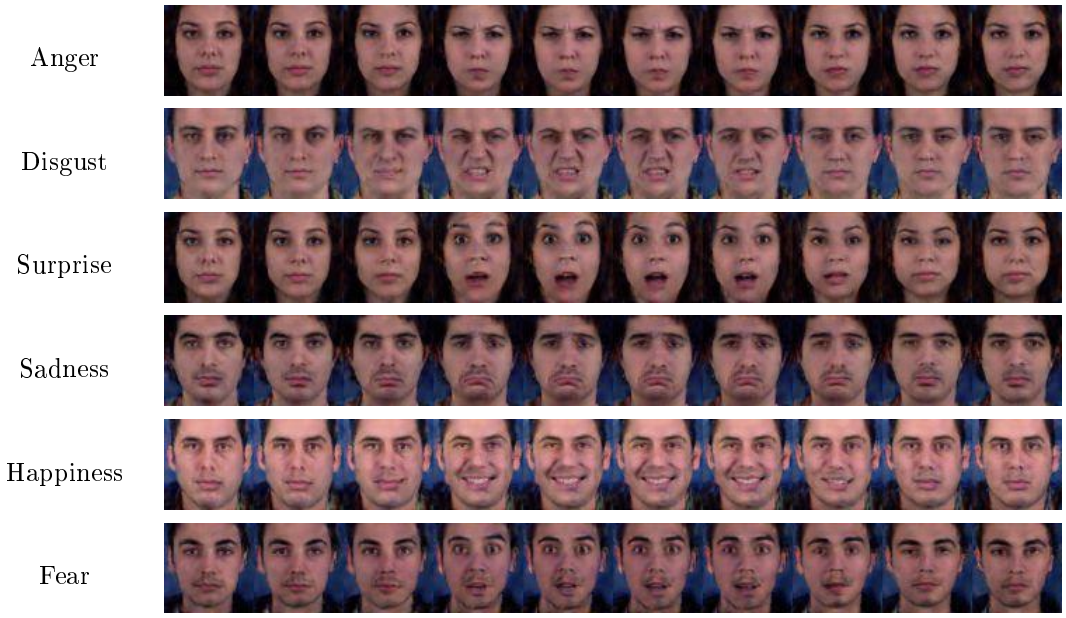


Figure 5.10: Generated video sequences from the onlyEM model. Each row corresponds to a different emotion class, showing 10 uniformly sampled frames.

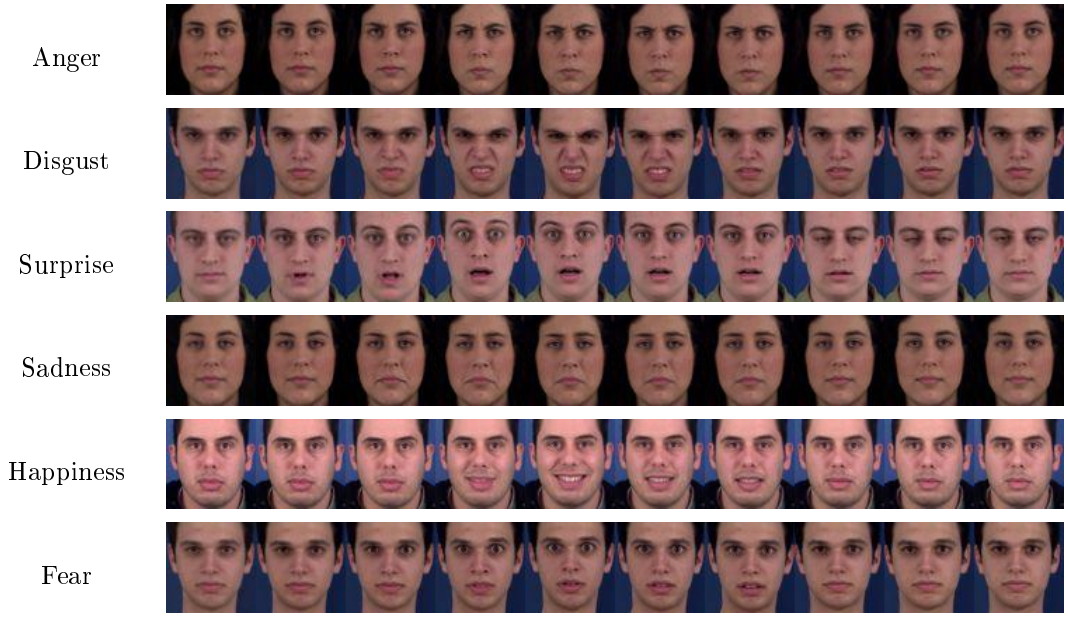


Figure 5.11: Generated video sequences from the onlyDM model. Each row corresponds to a different emotion class, showing 10 uniformly sampled frames.

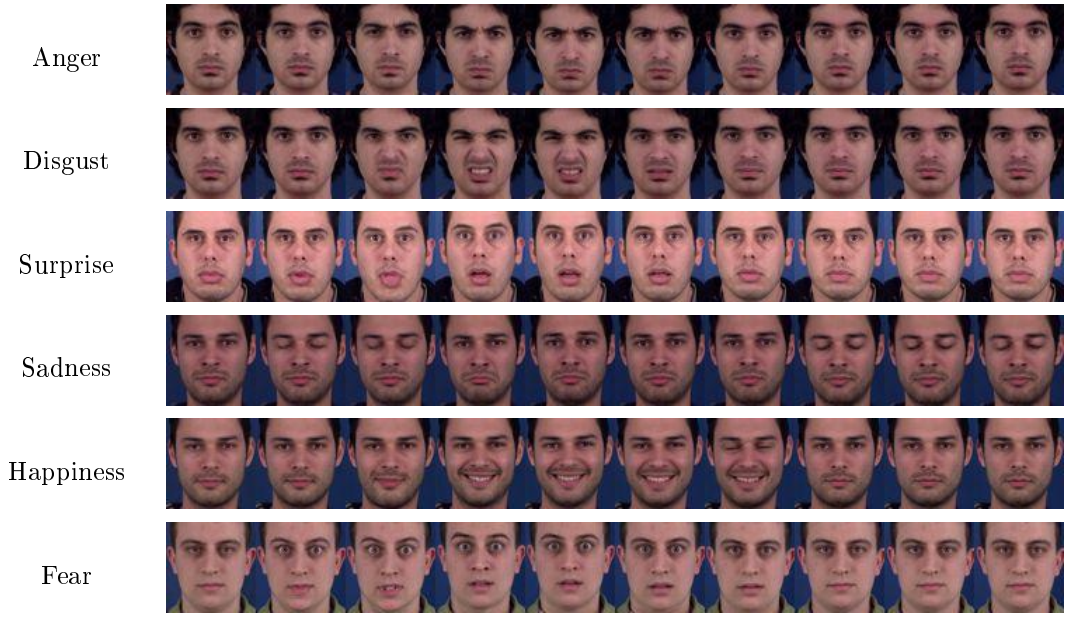


Figure 5.12: Generated video sequences from EM-DM (diffusion model retrained using the motion generator trained in the onlyEM setting, with explicit emotion conditioning). Each row corresponds to a different emotion class, showing 10 uniformly sampled frames.

Figure 5.13: Generated video sequences from both_retrievedMcs (diffusion model driven by motion codes retrieved from the training set using FECNet expression embeddings). Each row corresponds to a different emotion class, showing 10 uniformly sampled frames.

The visual comparison makes the qualitative gap between GAN-based and diffusion-based models immediately apparent. The VICEGAN **baseline** produces recognizable facial expressions, but the sequences can appear less sharp and often include visible artifacts or slightly unstable transitions. Moving from the baseline to **onlyEM** not only strengthens the perceived expression (consistent with its higher AEA/DEA), but also leads to a modest improvement in visual quality, with somewhat cleaner faces and fewer severe artifacts, even though the limitations of the GAN-based pipeline are still evident in many samples. In contrast, diffusion-based frame generation (**onlyDM** and **EM-DM**) produces markedly cleaner and sharper frames with stable dynamics and substantially fewer artifacts, in line with their much better FVD scores. Across all configurations, the temporal evolution from neutral to apex and back to neutral is generally plausible, indicating that each model learns meaningful emotional dynamics over time. This, in turn, highlights the need to go beyond automatic metrics and to incorporate human perception when evaluating how well emotions are conveyed.

5.2 Discussion

The results highlight a systematic trade-off between *perceptual quality* and *emotion recognition scores*, and clarify the role of each architectural modification.

Diffusion models improve overall video quality, with onlyDM as the best configuration. Across all settings, diffusion-based synthesis yields substantially lower FVD than GAN-based synthesis (Table 5.2). In particular, **onlyDM** achieves the best FVD, outperforming both the VICEGAN **baseline** and the SOTA reference **LFDM**; this is consistent with qualitative frames that appear cleaner and sharper.

Diffusion models preserve identity best. The ACD-I results show that diffusion-based variants achieve the strongest identity preservation across multiple embeddings, particularly under OpenFace and ArcFace (Table 5.4).

Temporal continuity is high for all models, with a small diffusion advantage. ACD-C recognition rates are high across configurations (often saturated under VGG-Face), but diffusion-based models tend to show slightly better distances and recognition rates (Table 5.5), indicating fewer short-term discontinuities.

GAN-based models score higher on emotion metrics, but this does not guarantee better realism. The best AEA/DEA values are obtained by GAN-based configurations, especially **onlyEM** (AEA = 0.76), and this trend is also visible when comparing against the VICEGAN **baseline**. A plausible reason is the presence of adversarial training components that explicitly reward outputs that are easy to classify as the target emotion. However, the simultaneous degradation in FVD for GAN-based models indicates that these gains may partially stem from distortions or artifacts that increase classifier confidence while reducing naturalness.

Conditioning shift improves emotion metrics and slightly improves FVD in the GAN setting, but not in the diffusion setting. Moving emotion conditioning to the motion generator improves emotion accuracy (AEA from 0.70 to 0.76) and reduces FVD (from 430 to 346) when the frame generator remains GAN-based (**baseline→onlyEM**). In contrast, when combined with the diffusion frame generator (**onlyDM→EM-DM**), both emotion metrics and FVD slightly decrease (AEA 0.64 → 0.58, FVD 100 → 120). This suggests that the diffusion model benefits from motion codes that are more stable and less variable during training, and that it can inject the desired temporal emotional

dynamics directly from the conditioning signal, without needing large changes in the motion-code distribution.

This interpretation is consistent with the behavior of **both_retrievedMcs**: using retrieved (real) motion codes can partially stabilize the motion trajectory at inference time, but it still does not match the best diffusion-only configuration in overall quality.

Automatic metrics have limitations and motivate human evaluation.

After a careful joint analysis of the quantitative results and the generated samples, it becomes clear that automatic metrics only partially capture how humans perceive the videos. Across several configurations, the numerical trends and the visual evidence do not always correspond perfectly. In particular, some models achieve higher emotion scores even when the generated faces exhibit visible artifacts, local deformations, or reduced naturalness.

This mismatch is especially evident for emotion-related metrics that rely on the outputs of external classifiers. Classifier confidence may increase even when perceptual quality does not improve and, in some cases, may even be driven by unnatural facial deformations or visual artifacts. For example, unnatural eye shapes may be interpreted as signals of surprise, dark pixel regions near the mouth may resemble cues associated with disgust, and unrealistically large, bright teeth within an exaggerated mouth shape may be interpreted as indicators of happiness.

By contrast, the metrics based on comparisons in feature space, such as FVD and the identity and continuity measures, tend to show a closer correspondence with the visual quality observed in the generated samples. Although these metrics are not a perfect proxy for human judgment either, in our analysis they are generally more consistent with perceived realism, facial stability, and temporal coherence than the classifier-based emotion scores.

Final model selection: onlyDM vs. EM-DM. The results consistently indicate that the diffusion-based variants **onlyDM** and **EM-DM** are the strongest overall models. Across the principal evaluation dimensions, they clearly outperform the remaining configurations: they obtain the best FVD scores, the strongest identity-preservation results, and the most stable temporal-continuity values, while also producing visibly cleaner and more coherent samples. Although some GAN-based variants obtain higher emotion-classification scores, these gains are associated with more evident artifacts, local facial deformations, and a less natural overall appearance.

To support the final selection between these two models, we report below the aggregate (*all*) results extracted from Tables 5.2, 5.3, 5.4, and 5.5, together with the aggregated AEA/ACD-I temporal profiles.

Model	FVD↓	DEA↑	AEA↑	DEA WN↑	AEA WN↑
onlyDM	**100**	**0.66**	**0.64**	0.46	0.49
EM-DM	120	0.63	0.58	**0.56**	**0.52**

Table 5.6: Aggregate (*all*) FVD and emotion metrics for **onlyDM** and **EM-DM**. Lower is better for FVD, higher is better for DEA/AEA and their WN variants. Values reproduce the corresponding entries in Tables 5.2 and 5.3.

Model	OpenFace (ACD-I)				ArcFace (ACD-I)				VGG-Face (ACD-I)			
	Cos_d↓	Cos_rr↑	Euc_d↓	Euc_rr↑	Cos_d↓	Cos_rr↑	Euc_d↓	Euc_rr↑	Cos_d↓	Cos_rr↑	Euc_d↓	Euc_rr↑
onlyDM	**0.08**	0.72	0.36	**0.87**	**0.16**	0.38	**2.58**	0.94	**0.08**	**1.00**	0.22	**1.00**
EM-DM	**0.08**	**0.73**	**0.35**	0.86	**0.16**	**0.39**	2.60	0.94	**0.08**	**1.00**	**0.21**	**1.00**

Table 5.7: Aggregate (*all*) identity preservation results (ACD-I) for **onlyDM** and **EM-DM**. Lower distance and higher recognition rate indicate better identity preservation. Values reproduce the corresponding entries in Table 5.4.

Model	OpenFace (ACD-C)				ArcFace (ACD-C)				VGG-Face (ACD-C)			
	Cos_d↓	Cos_rr↑	Euc_d↓	Euc_rr↑	Cos_d↓	Cos_rr↑	Euc_d↓	Euc_rr↑	Cos_d↓	Cos_rr↑	Euc_d↓	Euc_rr↑
onlyDM	**0.03**	**0.94**	**0.23**	**0.98**	**0.08**	**0.76**	**1.76**	1.00	**0.03**	**1.00**	0.14	**1.00**
EM-DM	0.04	0.93	0.24	0.97	**0.08**	0.72	1.84	1.00	0.04	**1.00**	**0.14**	**1.00**

Table 5.8: Aggregate (*all*) temporal continuity results (ACD-C) for **onlyDM** and **EM-DM**. Lower distance and higher recognition rate indicate smoother transitions. Values reproduce the corresponding entries in Table 5.5.

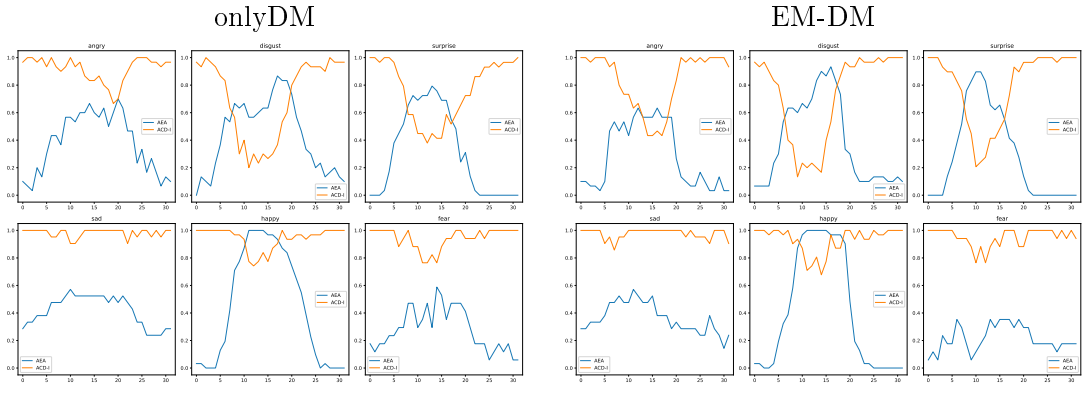


Figure 5.14: AEA and ACD-I temporal profiles for **onlyDM** and **EM-DM** (for each emotion).

In summary, **onlyDM** is selected as the final model. It achieves the best aggregate FVD, confirming the strongest overall generation quality, and it is equal or

slightly better than **EM-DM** on most aggregate ACD-I and ACD-C indicators, showing stronger identity preservation and temporal stability. This conclusion is reinforced by the aggregated temporal profiles in Figure 5.14, which show a slightly more stable identity trend and a cleaner evolution of the expression over time.

The emotion metrics do not overturn this conclusion. Although **EM-DM** is slightly better on the WN variants, **onlyDM** remains stronger on the standard aggregate DEA and AEA measures, and, more importantly, its qualitative samples are cleaner and less affected by visible artifacts, local deformations, or manipulations of facial structure. In other words, the small WN advantage of **EM-DM** is not accompanied by a stronger overall visual result, whereas **onlyDM** remains superior on the main quality indicators and on the perceptual appearance of the generated faces. For these reasons, **onlyDM** is the strongest overall configuration and is therefore chosen as the final model.

5.3 User Study: Human Evaluation of Generated Videos

As discussed in the paragraph on the limitations of automatic metrics, scores such as FVD, AEA/DEA and ACD-I/ACD-C are useful for large-scale benchmarking, but they do not directly measure how humans perceive the generated videos. In line with this view, several works on video generation [14, 30] argue that automatic metrics should be complemented by structured human evaluation to obtain assessments that are more reliable and better aligned with human perception. Therefore, to complement our quantitative analysis, we conducted a user study in which the selected diffusion-based model (**onlyDM**) was compared with both the **VICEGAN** baseline and the reference state-of-the-art model (**LFDM**), so as to evaluate the effective improvement achieved by the proposed work with respect to both the original baseline and the main external reference.

5.3.1 Protocol

To facilitate the user study, we developed a simple online platform where participants could view and compare the generated videos. For each of the six basic emotions (Anger, Disgust, Fear, Happiness, Sadness, Surprise), the platform si-

multaneously displayed three anonymized video clips, one from each model under comparison. The videos were presented as looped, 32-frame sequences at a resolution of 64×64 pixels. The order of the models was randomized for each participant and for each emotion to prevent ordering bias.

Participants were asked to rate each of the three videos for each emotion based on four quality dimensions, assigning a score from 1 (very bad) to 5 (excellent).

Based on the quantitative results from the previous sections, we evaluated three models: **VICEGAN** (the baseline), **LFDM** (the state of the art), and **onlyDM** (our best-performing model). In this way, we directly compare our best model against both the baseline and the state of the art.

Participants rated the videos according to the following criteria, on a scale from 1 (very bad) to 5 (excellent):

1. **Identity Preservation**: "Is the identity preserved?"
2. **Identity Consistency**: "Does the identity remain consistent throughout the video?"
3. **Emotion Generation**: "Is the desired emotion generated correctly?"
4. **Realism**: "Does the video appear realistic?"

5.3.2 Results

The study followed a within-subjects design, where each of the **93** participants provided 72 ratings in total (one for each of the 3 models across 4 criteria and 6 emotions) for a total of 6696 ratings. For a given criterion and emotion, each participant assigned an integer score from 1 (very bad) to 5 (excellent) to every model. We then computed the **mean score** for each model by averaging its scores across all participants and emotions. The resulting values lie in the range $[1, 5]$, where a higher mean score indicates a stronger preference for that model.

Model	Mean score	Ratings
VICEGAN	2.60	2232
LFDM	3.80	2232
onlyDM	**4.09**	2232

Table 5.9: User study results: overall mean score (higher is better, on a scale of 1–5) and number of ratings for each model across all criteria and emotions.

Model	Identity Pres.	Identity Cons.	Emotion	Realism
VICEGAN	2.48	2.35	3.31	2.21
LFDM	3.96	4.02	3.33	3.87
onlyDM	**4.14**	**4.12**	**4.10**	**4.04**

Table 5.10: User study results: mean score (higher is better, on a scale of 1–5) for each model across the four evaluation criteria. Scores are averaged over all participants and all six emotions.

Model	Anger	Disgust	Fear	Happiness	Sadness	Surprise
VICEGAN	2.72	2.89	2.37	2.40	2.57	2.63
LFDM	3.70	**3.88**	3.50	3.99	3.58	4.17
onlyDM	**4.14**	3.66	**4.16**	**4.34**	**3.98**	**4.28**

Table 5.11: User study results: mean score (higher is better, on a scale of 1–5) for each model and each emotion. Scores are averaged over all participants.

To determine if the observed differences in ratings are statistically significant, we performed a Friedman test for each criterion. A significant result ($p < .05$) suggests that participants did not rate the models equivalently, meaning at least one model was consistently rated higher or lower than the others.

Criterion	$\chi^2(2)$	p
Identity Preservation	90.738	1.98×10^{-20}
Identity Consistency	91.959	1.07×10^{-20}
Emotion Generation	65.136	7.18×10^{-15}
Realism	85.549	2.65×10^{-19}

Table 5.12: Friedman test statistics for each evaluation criterion in the user study.

Overall, the user study indicates that **onlyDM** is the most preferred model, clearly outperforming both **LFDM** and the **VICEGAN baseline** in terms of mean score.

The statistical analysis confirms that differences between models are highly significant across all evaluated criteria, namely identity preservation and within-video identity consistency, correctness of emotion generation, and video realism. Moreover, participants showed a moderate and stable level of agreement in their judgments for each criterion.

Chapter 6

Conclusions

6.1 Analysis of Findings

This thesis investigates the generation of short facial-expression videos from a single neutral image and a target emotion, with a focus on realism, identity preservation across the sequence, and believable expression dynamics. Building on the experiments reported in Chapter 5, this section revisits the research questions introduced in Chapter 1 and discusses them in order (RQ1–RQ4).

In the following, we use the short model names introduced in Chapter 3: the VICEGAN **baseline**, the motion-conditioned variant **onlyEM**, the diffusion-based frame generator **onlyDM**, their combination **EM-DM**, and the retrieval-based variant **both_retrievedMcs**.

6.1.1 RQ1: Conditioning strategy for emotion-aware motion representations

RQ1 asks how the conditioning strategy can be designed so that motion codes acquire emotion-related semantics and yield emotion-specific temporal dynamics. In this thesis, we answer it by moving the emotion signal from the frame generator to the motion generator (**onlyEM**), so that the motion trajectory itself carries the expression dynamics.

In this setting, the frame generator no longer receives an explicit emotion label and must infer the target expression from the motion trajectory it is conditioned on. As a result, the motion generator is pushed to encode emotion-specific tem-

poral dynamics into the motion codes, since they become the only carrier of expression information available to the frame synthesizer.

6.1.2 RQ2: Replacing the GAN with a diffusion model

RQ2 asks whether diffusion models can provide superior frame synthesis quality compared to GAN-based approaches. The results provide a clear positive answer: replacing the GAN frame synthesizer with a diffusion model (**onlyDM**) yields the strongest overall improvements in realism, identity preservation, and temporal continuity.

Diffusion-based synthesis strongly reduces FVD and improves ACD-I and ACD-C, meaning that frames are closer to real videos, identity is better preserved, and frame-to-frame continuity is smoother. Visual examples confirm these gains: **onlyDM** produces sharper faces, fewer artifacts, and more stable identities at the emotion apex.

For emotion metrics, diffusion-based variants obtain lower AEA/DEA than the best GAN-based configuration. However, as explained in Chapter 5, these scores depend on pretrained emotion classifiers and can be increased by artifacts or exaggerated expressions that do not look realistic. The user study helps reconcile these perspectives: participants consistently preferred onlyDM over both the VICEGAN baseline and the LFDM reference across identity, emotion, and realism questions, supporting the selection of onlyDM as the final model.

6.1.3 RQ3: Leveraging pretrained emotion embeddings for video generation

RQ3 asks whether a pretrained model whose embeddings already capture emotion semantics can drive facial-expression video generation, transferring expression dynamics from the training set to unseen identities. The **both_retrievedMcs** configuration tests this idea: motion codes are not generated by an RNN but retrieved from real training sequences by matching FECNet expression embeddings, and the diffusion model synthesizes frames conditioned on these retrieved trajectories.

This design has two practical advantages. First, the retrieved motion codes inherit realistic temporal dynamics directly from real data. Second, there is no need to train a motion generator: the emotion trajectories are extracted with a pretrained

network, which simplifies the training pipeline to a single diffusion stage.

The quantitative results are mixed. On the positive side, **both_retrievedMcs** achieves an FVD of 287, improving over both the VICEGAN baseline (430) and LFDM (291), and it outperforms both on several identity metrics (ACD-I). This confirms that the pretrained embeddings do transfer meaningful expression information to unseen identities. On the negative side, the temporal AEA profiles show that **both_retrievedMcs** does not produce the expected emotion peak around the apex frames and exhibits higher frame-to-frame variability than the other diffusion variants. Its ACD-C scores also drop significantly, indicating less stable transitions. Overall, the variant falls clearly short of **onlyDM** (FVD=100) in both distributional quality and temporal coherence.

The reason is that the retrieved motion codes, being copies of real trajectories, carry more complex and variable dynamics than the motion codes learned by the RNN. The current diffusion conditioning interface is unable to handle this variability: it works best with smoother, lower-variance motion inputs.

6.1.4 RQ4: How motion-code semantics influence final video quality

RQ4 asks how the semantic content of the motion codes influences the quality of the generated videos.

The clearest evidence comes from comparing configurations that differ only in what the motion codes carry. In the GAN setting, adding emotion semantics helps: **onlyEM** raises AEA from 0.70 to 0.76 and reduces FVD from 430 to 346 relative to the **baseline**, showing that richer motion codes translate into better emotion accuracy and distributional quality.

In the diffusion setting, the opposite happens. **EM-DM**, which feeds emotion-conditioned motion codes to the denoising network, degrades FVD from 100 to 120 and AEA from 0.64 to 0.58 compared to **onlyDM**, which uses emotion-neutral ones. Temporal continuity also worsens slightly. The pattern is consistent: whenever the motion codes become more variable, whether because they carry emotion information (EM-DM) or come from real sequences (both_retrievedMcs, as discussed in RQ3), the diffusion model produces worse results.

This asymmetry reveals that the bottleneck is not the motion-code content itself, but how the denoising network receives it. The current cross-attention mechanism

works only when temporal embeddings are stable and low-variance, with the emotion signal injected separately at the frame-generation stage. Richer semantics are not inherently harmful (they clearly help the GAN), but the denoising network cannot exploit them because it fails to disentangle the additional variability from the useful temporal structure. Redesigning this component is therefore a necessary step to unlock the full potential of semantically richer motion codes in diffusion-based pipelines.

6.2 Contributions and Innovations

The main contributions of this thesis are:

- **A comparison of two key design choices** for emotional video generation: (i) where to inject emotion conditioning (in the motion codes or in the frame generator), and (ii) which frame generator to use (GAN or diffusion).
- **An emotion-aware motion-code generator** that applies conditioning before frame synthesis and produces more convincing emotional dynamics than the VICEGAN baseline.
- **A diffusion-based frame generation pipeline** that significantly improves realism, identity preservation, and temporal stability, resulting in the best overall model (**onlyDM**).
- **An evaluation protocol combining automatic metrics and a user study**, including video-level realism (FVD), emotion scores (AEA/DEA), identity and continuity metrics (ACD-I/ACD-C), temporal trend analysis, and human judgments.
- **A decoupled training framework** that separates motion extraction and frame synthesis, using an external network to extract semantic emotion representations and showing how the components interact.

6.3 Limitations and Challenges

This work also has limitations that should be considered when interpreting the results.

- **Resolution and dataset scope.** Experiments are conducted at 64 × 64 resolution and on a single dataset (MUG). This setting is useful for controlled comparisons, but it limits conclusions about high-resolution and in-the-wild data.
- **Emotion metrics are imperfect.** AEA/DEA depend on a pretrained emotion classifier. These scores are informative, but they can be biased and they do not directly measure human perception.
- **Compute cost.** Diffusion sampling is significantly slower than GAN inference. This is a practical challenge for real-time or large-scale deployment.
- **Component mismatch.** The combined EM-DM configuration indicates that improved motion-code semantics do not automatically transfer into better diffusion-based videos. This exposes an interface problem that requires further research.

6.4 Future Work and Extensions

The findings of this thesis suggest several concrete directions for future research.

- **Build on the retrieval-based pipeline and redesign the conditioning mechanism.** The **both_retrievedMcs** configuration provides a clean and scalable setup where motion dynamics are imported from real sequences via motion-code retrieval while frame synthesis is learned in a single diffusion training stage. However, the analysis in RQ3 and RQ4 shows that the current conditioning interface struggles to handle variable or semantically rich motion codes. Future work should therefore explore alternative conditioning strategies that allow the diffusion model to better exploit richer motion representations.
- **Move to higher resolution and more diverse data.** Extending the pipeline to higher-resolution faces and more varied datasets would test robustness and practical value.
- **Use better metrics.** Improved perceptual metrics can provide more reliable conclusions than classifier-based scores alone.
- **Faster sampling.** More efficient samplers and lightweight diffusion processes (e.g., DDIM) can reduce inference time while preserving quality.

List of Figures

List of Tables

Bibliography

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation (OSDI)*, pages 265–283, 2016.

[2] N. Aifanti, Christos Papachristou, and Anastasios Delopoulos. The MUG facial expression database. In *Proc. 11th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, pages 1–4, 05 2010.

[3] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. In *CMU School of Computer Science Technical Report*, 2016.

[4] Greco Antonio, Strisciuglio Nicola, and Vento Mario. Animating faces with emotions through a generative adversarial network preserving identity. *IEEE Transactions on Affective Computing*, page 1–12, 2025.

[5] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6299–6308, 2017.

[7] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In D. Lee, M. Sugiyama, U. Luxburg,

I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[8] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[9] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *CoRR*, abs/2105.05233, 2021.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020.

[11] Paul Ekman. An argument for basic emotions. *Cognition and Emotion*, 6(3-4):169–200, 1992.

[12] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[13] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández Del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020.

[14] Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhranil Chandra, Ziyan Jiang, Aaran Arulraj, Kai Wang, Quy Duc Do, Yuansheng Ni, Bohan Lyu, Yaswanth Narsupalli, Rongqi Fan, Zhiheng Lyu, Yuchen Lin, and Wenhu Chen. Videoscore: Building automatic metrics to simulate fine-grained human feedback for video generation, 2024.

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020.

[16] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022.

[17] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models, 2022.

[18] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.

[19] John D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[20] Haomiao Ni, Changhao Shi, Kai Li, Sharon X Huang, and Martin Renqiang Min. Conditional image-to-video generation with latent flow diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18444–18455, 2023.

[21] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015.

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[23] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer, 2015.

[25] Sefik Ilkin Serengil and Alper Ozpinar. Lightface: A hybrid deep face recognition framework. In *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pages 23–27. IEEE, 2020.

[26] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers,

Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision, 2021.

[27] Raviteja Vemulapalli and Aseem Agarwala. A compact embedding for facial expression similarity, 2019.

[28] Jing Xiong, Gongye Liu, Lun Huang, Chengyue Wu, Taiqiang Wu, Yao Mu, Yuan Yao, Hui Shen, Zhongwei Wan, Jinfa Huang, et al. Autoregressive models in vision: A survey. *arXiv preprint arXiv:2411.05902*, 2024.

[29] Zhiyu Yin, Kehai Chen, Xuefeng Bai, Ruili Jiang, Juntao Li, Hongdong Li, Jin Liu, Yang Xiang, Jun Yu, and Min Zhang. A survey: Spatiotemporal consistency in video generation. *arXiv preprint arXiv:2502.17863*, 2025.

[30] Tianle Zhang, Langtian Ma, Yuchen Yan, Yuchen Zhang, Kai Wang, Yue Yang, Ziyao Guo, Wenqi Shao, Yang You, Yu Qiao, Ping Luo, and Kaipeng Zhang. Rethinking human evaluation protocol for text-to-video models: Enhancing reliability, reproducibility, and practicality, 2024.