National College of Ireland

Investigating SOTA Deepfake Detection frameworks in content-driven multimodal deepfake content

MSc Research Project
MSc Artificial Intelligence

Abhishek Gupte
Student ID: 22183604

School of Computing
National College of Ireland

Supervisor: Anu Sahni

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Abhishek Gupte
Student ID:	22183604
Programme:	MSc Artificial Intelligence
Year:	2025-2026
Module:	MSc Research Project
Supervisor:	Anu Sahni
Submission Due Date:	23/12/2025
Project Title:	Investigating SOTA Deepfake Detection frameworks in content-driven multimodal deepfake content
Word Count:	5917
Page Count:	29

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	30th December 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Investigating SOTA Deepfake Detection frameworks in content-driven multimodal deepfake content

Abhishek Gupte
22183604

Abstract

Deepfake generation technology has evolved over the past several years in unprecedented ways. A majority of such media are either entirely fake or entirely real & are reasonably easy to identify using current state-of-the-art binary-level classification deepfake detection systems. However, in recent years a new form of deepfake temporal forgery dataset, called the LAV-DF has emerged. This consists of deepfake videos wheres only a few temporal regions have been manipulated to change the sentiment/meaning of the entire video has emerged. Current traditional deepfake detection architectures need severe modification in order to keep up with the accuracy of detecting temporal localised datasets, which require a compromise in terms of environmental efficiency. This work has focused on implementing a version of a popular SOTA multimodal deepfake detection architecture called the NPVForensics - trained on the LAV-DF, evaluating weaknesses in the architecture and point to future directions. The results have shown the SOTA architecture under constrained resources has not performed well on the dataset. The future directions propose implementation and fine-tuning approaches to mitigate difficulties.

1 Introduction

As AI technology becomes increasingly sophisticated, so does the looming threat of facial forgery. Deepfake generation, which started as a facet for entertainment, became increasingly malicious following the infamous Reddit controversy in 2017 when fake pornographic material of non-consenting celebrities was released. The threat continues to grow with more realistic deepfake. The hyper-realism in recent synthesized videos make it it hard to distinguish from their real-life counterparts. At it's most extreme, its touted to bring about an epistemological crises , where the public loses faith in authority figures because of the widespread fake news propagated through deepfake scenarios (Habgood-Coote; 2023).

Several efforts have been done in AI technology to counter the threat of Deepfake Detection. A majority of conventional work on deepfake detection until recent years was focused on detecting visual manipulations. These approaches started originally with the utilization of ML frameworks, which eventually became obsolete. due to their inefficiency capture fine-grained temporal features (Weerawardana and Fernando; 2021). As deep learning technologies began to grow, the ability to detect fine-grained information due to lack of temporal information. Hence, researches began to use CNN-based architectures coupled with memory networks such as LSTMs to detect temporal consistencies within

the continuity of the frames (Alrashoud; 2025). Simultaneously work began on audio deepfakes which utilised spectrogram generation to find audio mismatches between the source and target speaker.

Multi-modal frameworks , are set to be more effective than traditional single modality networks by utilising the inter-relationships between audio and visual modality simultaneously (Wang; 2025).

However, since most deepfakes involve binary classification of 'Real' or 'Fake', the duration of the entire video is input to the audio/video networks. Off-late, an advanced & emerging form of content-driven manipulation media has complicated the issue of detection. These form of deepfakes are designed such that only a tiny portion of the video is changed, thereby altering the entire meaning of the statement uttered by the speakers. This particular class of datasets are challenging traditional unimodal and multimodal frameworks requiring a new adaptation or several modifications. (Cai et al.; 2023a). They utilise a Boundary Detection technique h alongside 3D-CNNs to enhance temporal localization to enable detection of deepfakes in those videos. These techniques contain stacks of models with intensive training needs & put a setback to the fair usage of green energy.
This work will investigate an existing state-of-the-art framework done on this area on a content driven manipulated deepfake dataset. In order to meet a middle ground on the usage of resources, the module has undergone several modifications and will be utilised in a resource-constrained & productive environment.

1.1 Research Objective(s)

The broader research goal is to examine the performance of existing near-optimal deepfake multi-modal frameworks on a newer content-driven dataset motivated by temporal localisation in a moderately resource-constrained environment. The research question is described as -

"How effective and under what metrics can a partially implemented state-of-the-art work done in the domain of traditional multimodal deepfake detection be assessed on a broad out-of-domain temporal-forgery dataset?"

2 Related Work

Several methods are used to detect deepfakes from CNNs due to their superiority to learn fine-graned local features, LSTMs, using Person of Interest (POI) bio-markers, GANs and transformer based mechanisms that conduct cross-modal learning (Weerawardana and Fernando; 2021). A notable work by (Lewis et al.; 2020) uses a tri-modular network which learns spatial features, their temporal dynamics as well as conducts experiments with frequency-based spectral features. They utilise a tri-network setting where each modality is enhanced without using middle or cross-fusion techniques.

2.1 Cross-modal attention mechanisms

The survey work done by (Wang; 2025) reviews multimodal deepfake work and enlists their usages into early, middle and late fusion mechanisms. They highlight the importance

of attention-based fusion, contrastive learning, and representation alignment learning. They highlight the importance of inter-fusion learning about when audio and video to be pulled together when they originate from the same video. The type of learning with an InfoNCE-based loss ensures weighted cross-modal learning to enhance model predictions.

2.1.1 Swin Transformers

The Swin Transformers (Liu, Lin, Cao, Hu, Wei, Zhang, Lin and Guo; 2021) utilize a shifted-attention window in order to read patches within the images. The novelty lies in their shifted-attention window to capture both both local & global temporal information within the image patches. This is achieved with the help of a shifted window policy, which ensures localised self-dependence within the patches and global exchange between patches in the image. The Swin Transformer does not typically require pre-tokenized embeddings, as would be the case in its predecessor the Visual Transformer (ViT) Chung et al. (2018) since this information is already obtained using a patch embedding layer (Dosovitskiy; 2020).
In comparision, the Video Swin Transformer (Liu, Ning, Cao, Wei, Zhang, Lin and Hu; 2021a,b) performs the same task but with the additional temporal component spread out amongst frames in sequence such as videos. It creates 3D-patch tublets making it efficient to process videos, offering a computational tradeoff in exchange of modifying existing Swin architecture to operate on video sequences.
Swin Transformers have been used extensively in the multimodal deepfake literature. AVFakeNet (Ilyas et al.; 2023) entirely utilize the concept of Swin Transformers in the Feature Extraction Block. They modify the original Swin architecture by adding an extra Layer Normalisation stage, and by changing the Window-Attention policy from Shifted-Window from Window-based attention between the first and second Swin blocks. This allows the leveraging of cross-window features from the second Swin block downwards. The same dense Swin network has been used to model audio & video after passing them through a primary dense feature generation block. As the modality shapes are different, a modality tokeniser is an essential step for the model to generalise both modalities on the same scale.

2.2 Self-Supervised Learning

Ths is a type of unsupervised learning mechanism which is suitable for usage when data labels are not available for models to learn in real world situations. The work by (Gui et al.; 2024) surveys SSL and describes tasks that require learning discriminative, clear features that can be used for a wide range of tasks from similarity/dissimilarity learning, correlation analysis etc as well as representation learning in videos. These strategies have been used in Deepfake detection by establishing the authenticity of facial forgery videos, utilizing the discrepancies between facial feature & audio feature classification.

2.3 NPVForensics - A novel architecture in multimodal deepfake

The work done by (Chen et al.; 2025) utilises a two-stage pipeline - a self-supervised pre-taining stage on real videos & a later fine-tuning stage on the target deepfake dataset. The basis of subject detection are phoneme-viseme pairs, which correlate spoken speech

with the visual act of speaking. The top model utilising every stage has a supreme 99.9% performance on the FF+ dataset. They utilise a feature extraction, common projection, feature alignment module to compute audio-visual multimodal learning. The stages used have been described in later sections.

2.4 Energy monitoring in Deep Learning techniques

(Tmamna et al.; 2024) describe architectures used in deep learning as consuming vast amounts of energy. They recommend optimizing the architecture (shrinking models by removing unnecessary weights & layers) as an important technique in order to sustainably utilize energy. It's also important to trade off energy & efficiency along with using data selection. One of the key points was deciding how much data to keep depending on the utility of the model. They also asserted the usage of memory-efficient architecture such as NVIDIA's specialised GPU cores which would reduce the overall footprint used in training models.

2.5 Datasets

2.5.1 LAV-DF

The original LAV-DF dataset was created by (Cai et al.; 2023b; **?**) to model deepfake challenges where only a small portion of the video is modified to alter the entire meaning. Amongst creating a end-to-end frame classification pipeline, the work also utilises a Frame Boundary-Aware algorithm similar to (Su et al.; 2021) which generates boundary proposals to localize apparent fake segments within a video. This involves creating a large matrix where the model consumes guidance on at frame-transitions could the fake segments be embedded into the video. Temporal localisatio is essential to determine the fake segment, as a very large portion of the video is not tampered with.

The data set labels distinguish distinctly between fake audio, fake video, real audio & real video. It contains more than 136,000 videos in total, out of which approximately 99,000 have been manipulated.

2.5.2 AVSpeech

This audio-visual dataset was originally built for detection of acoustics and visual signals containing labeled high quality Youtube videos with several speakers speaking in various dialects and languages. It contains up-to 470,000 video clips with nearly several hundreds of thousands of unique speakers. (Ephrat et al.; 2018)

2.5.3 DFDC

A very popular audio-visual deepfake dataset which is generated using FaceSwap. It consists of a large high-variance of speakers in diverse poses in substantial spread of settings. It is one of the largest datasets released for deepfake detection with over 470k videos used. It consists of a total of 124k videos out of which approximately 20k videos are real (Dolhansky et al.; 2020; Brian Dolhansky; 2020).

3 Methodology

This section will cover the end-to-end pipeline, model architecture, outline key design stages and solutions.

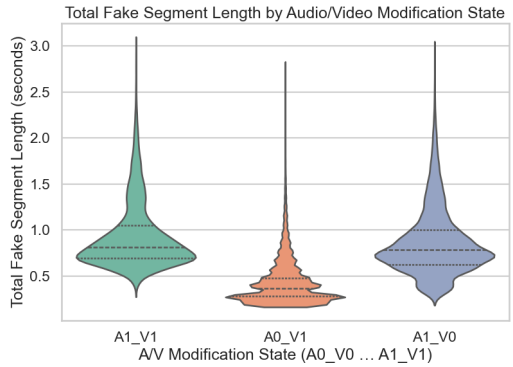
3.1 Execution Roadmap

These are summarised in sequence as -

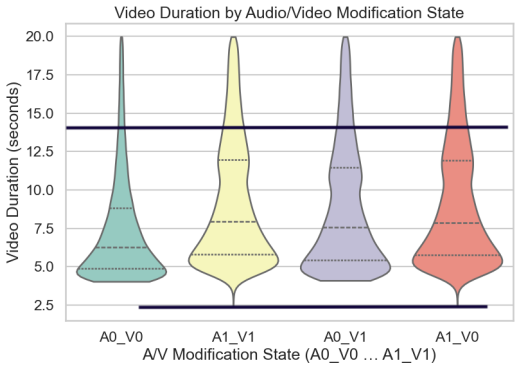
- Dataset Analysis - The section shall describe the analysis & overview all the datasets used in the work in the training & evaluation pipelines
- Preprocessing - The stages of preprocessing each segment within the audio & video modalities will be described
- Feature Extraction - The sub-methodologies used in feature extraction of both audio and video modalities along with auxillary operations post-feature extraction are explained here
- Model Development - The implementation of the original NPVForensics model are covered in detail along with the modifications and divergences
- Model Training - The original stages of Self-Supervised learning (Stage 1 model) & fine-tuning (Stage 2 architecture) are explained while highlighting motivation.
- Model Evaluation - The metrics used to evaluate each stage of the model, including energy consumption metrics

3.1.1 Dataset Analysis

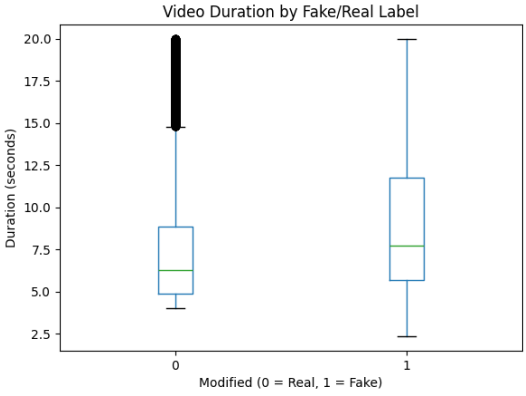
LAV-DF was pre-analysed to determine a niche area of focus to conduct model training & evaluation. The scope of study included combinations of audio/video segments, the duration of each clip & the length of fake segments within the videos. Plots were generated for each of the cases.



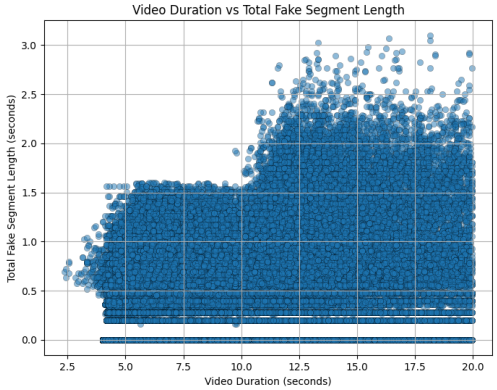
(a) Fake segment length vs A/V combinations



(b) Video duration vs A/V combinations



(c) Video duration (boxplot)



(d) Fake length vs video duration (scatter)

Figure 1: LAV-DF: relationships between fake segment lengths, audio/video availability combinations, and video duration.

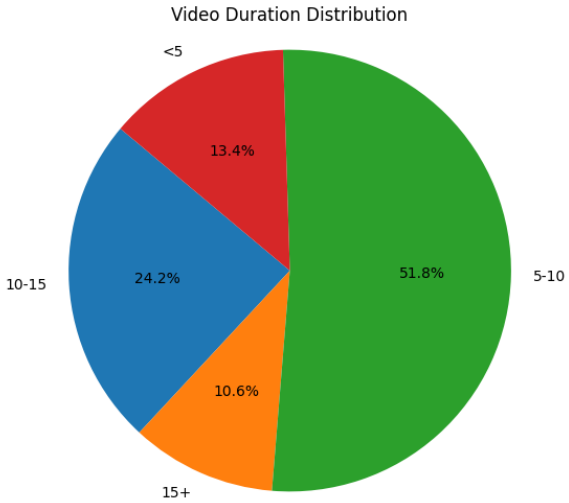


Figure 2: Video Duration Share (amongst all fake videos)

1. The scatter plot in figure 1d shows distribution of samples corresponding to fake

segment lengths across video durations. Either of the modalities are fake in this scenario. When both modalities are fake, the length of the fake segments are added together as a sum. We can observe there are only a few outlier segments when the length of either or both fake segments is the maximum range in the dataset (3 seconds). These also coincide with the videos that have the maximum length. We also observe, for the video duration range (5.0 - 10.0 seconds) the length of fake segments is a constant 1.5 seconds. Apart from this, the length of video duration,is almost proportional to the length of fake segments - the longer the video the longer the fake segments. Subsequently, in the next figure we observe that this range (5-10 seconds) consists of the highest proportion (more than than 50%)of total fake videos.

2. The box-plot in 1c compares side-by-side the quartile ranges of fake video segments with that of real videos. It can be directly inferred real (modified = 0) videos have more outliers (length of videos is greater than 15), whereas there is more of an even spread for the fake videos. The 3rd & 4th quartile groups of both are more spread out past 7.5 seconds.

3. A violin chart in 1b represents each modality combo (A signifies Audio; V signifies Video), and [0 - REAL ,1 - FAKE] represent the genuinity of videos. The spread of number of samples in the dataset are equivalent with a few minor differences between each of them. It can be seen that these are very few samples (approximately 2.5 seconds of total video length). These are taken up by the modality combo [Fake Audio,Fake Video] & [Fake Audio, Real Video], while the distribution remains near identical within comparision of each category.

The video duration of interest has been determined to be 7.5 seconds. This is the mean of the [5-10 second] range which contains the maximum number of rich training samples, and thus shall have variety in terms of enriched diversity of Fake segments. Extrapolating from the box-plot diagram , the median for video duration marks at 7.5 seconds. Hence the videos are chosen for training, testing & validation at or under 7.5 seconds.

3.1.2 AVSpeech Dataset

The dataset is downloaded using an online repository which crawls YouTube to download videos from the link-set corresponding to AVSpeech. Each video length is capped are capped at 7 seconds, in order to keep the number of frames in parity with the LAV-DF dataset, since models with equal length sequences are able to generaise better with fixed-length videos. The dataset consists of speakers of diverse backgrounds, speaking several languages. They are also of a higher resolution than the LAV-DF dataset videos. This dataset will primarily be used for pre-training the model architecture. The pretraining ensures model preparedness in the common projection and correlation learning blocks. The videos contain resolution of the highest quality of 640x380 amongst the sets of videos used in the work.

3.1.3 DFDC Dataset

The dataset contains non-speaker annotated videos spanning single-face segments where the spoken language is English. The sub-set of DFDC contains videos with both deepfake audio & video segments. However, they are not labeled and only a single label is assigned

to the video. The dataset contains both in-studio and in-the-wild video speakers, thereby ensuring a wide spread of visual and auditory signals. The dataset will be used both for training and testing.

3.2 Architecture Modeling

This section shall encapsulates the divergences & modifications from the work in NPV-Forensics, along with describing the stages outlined in Execution Roadmap.

3.2.1 Data Preprocessing

Component	**Original Work**	**Current Work**	**Remarks**
Video	Full-frame & lip segment (viseme); $V \in \mathbb{R}^{C\times T\times H\times W}$ $F \in \mathbb{R}^{C\times T\times H\times W}$	Single Full-Frame face crop $Face \in \mathbb{R}^{C\times T\times H\times W}$	Uses a single-stream face crop to integrate the idea of the full frame and lip segment into one facial region
Audio	A log Mel-spectogram with consisting of 64-mel features & sampled at 16,000 hz. $A \in \mathbb{R}^{F_m=64\times T=96}$	No change	The fidelity of audio in the NPVForensics pipeline was maintained

Table 1: Data Preprocessing

An overview of the data-preprocessing paradigm -

- NPVForensics utilises the concept of analysis based on phoneme-viseme pairs. The visemes (visual portion of speech) is characterised as the generation of lip features. These are clipped from the video during the "non-critical" phoneme sequences using timestamps from P2FA generated segments. The other video stream unit is a single frame, comprising of the scenery and the source/target speaker. A log-mel spectrogram is generated for the audio, as the third stream. All three of them are clipped from the same P2FA aligned phoneme feature-set.
  All three modalities are obtained from phoneme segments generated using an advanced algorithmic alignment technique based on P2FA. These boundaries are fine-grained with sensitive dynamics captured within the segment.

- The current work includes only one stream for video - the face sequence. This is seen as a low-energy workaround to two streams. The LAV-DF dataset is altered on temporal segments and does not require phoneme-viseme boundaries for operational detection. Moreover, the fine-grained phoneme pairs would be sensitive to temporal boundaries. The video is first transcribed to generate word-level transcriptions using a local installations of Whisper AI - transcription model [1]. Then the words-level segments are generated using a word-grouping algorithm. The face in the video is then detected, and the regions in the word-groups are cropped and stitched into the new video. Each word group corresponds to one "training segment".

[1]Found at Whisper AI GitHub page

3.2.2 Feature Extraction

The table highlights the differences between the original work and the sub-methodology followed in the current work.

Component	**Original Work**	**Current work**	**Remarks**
Video	Uses a modified Local-Feature Aggregation Swin-family based transformer. Prior to this a pool of per-modality tokenized embeddings are created to feed into the model; The model output is given as - $X_V \in \mathbb{R}_{\mathbb{V}}^{D_v \times S(\ tokenlength)}$	Uses the Video Swin Transformer ; The$X_V \in \mathbb{R}_{\mathbb{V}}^{C \times T \times H \times W}$	A more energy efficient solution to generating preserved spatio-temporal information across a sequence of frames
Audio	Uses the same Local-Feature Aggregation - Swin-family based transformer - Prior to this a separate per-stream tokenized positional embeddings are created to feed into the model; The model output is given as $X_V \in \mathbb{R}_{\mathbb{V}}^{D_A \times S(\ tokenlength)}$	Uses a vanilla swin-transformer after resizing the Audio input into a 1x224x224 image $R_V \in \mathbb{V}_{\mathbb{F}}^{C_f \times Tokens}$	The new setup does not assume pre-generated positional encodings.

Table 2: Feature Extraction

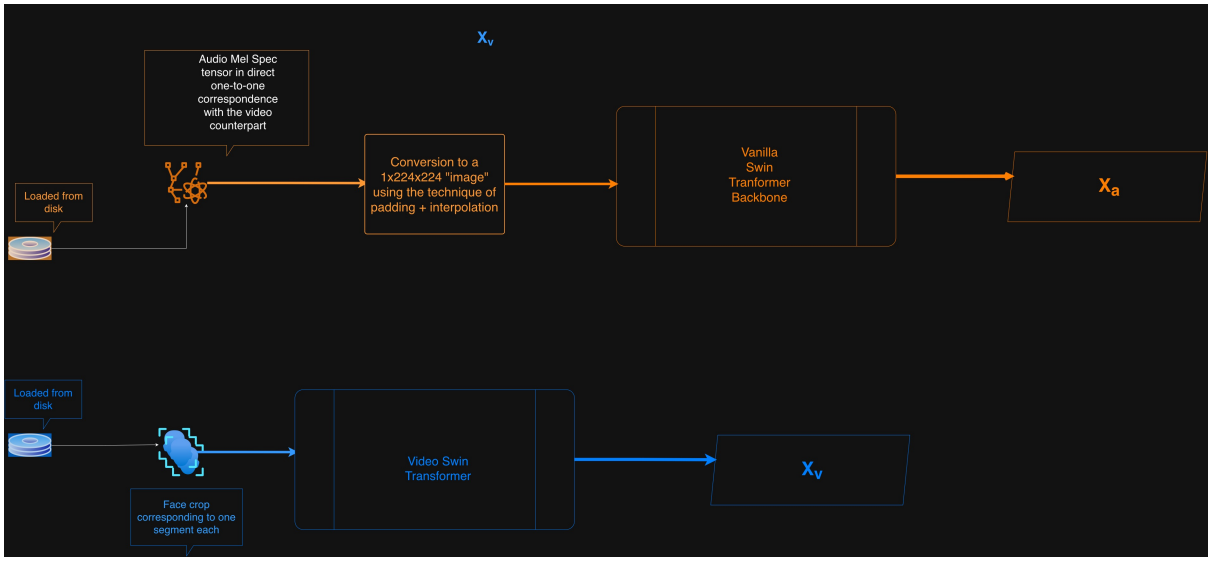


Figure 3: Feature Extraction sub-arch

3.2.3 Module A - Pre-alignment module

The first clear divergence from the original authors' work as it introduces a mean pooling layer to align the sequence lengths between the two output modalities from Swin transformers.

Original Work	Current Work	Remarks
The original work does not use a post-SWin tokenizer layer. It implements an absolute modality-tailored embedding layer prior to the task of feature extraction to temporally or spatially align audio and video pairs.	Follows the same paradigm, but with only one pair. The final loss is taken as the co-relation between the face frames and the audio spectrogram features.	Replicates the basic skeleton structure, but does not follow end-to-end as there are only two streams available

Table 3: Module A - Pre-alignment module

3.3 Module B - VACL Co-relation learning

The sub-section describes a working breakdown of the second sub-module - which is the architectural core of the NPVForensics paper. The table below diverges the idea behind the VACL block.

Original Work	Current Work	Remarks
A co-relation learning block conducts complex attention-weighted mechanisms involving two pairs of a total of three streams extracted from the LFA-SWIN Transformer; the two pairs are concatenated in the end to create the final discriminative features. The loss is generated to learn the symmetry patterns between each pair. Finally an Alignment Learning module is used to mine inter-modal consistency involving both pairs.	Follows the same paradigm, but with only one pair. The final loss is taken as the co-relation between the face frames and the audio spectrogram features.	Replicates the basic skeleton structure, but does not follow end-to-end as there are only two streams available

Table 4: Module B - VACL block

3.4 Model Architecture

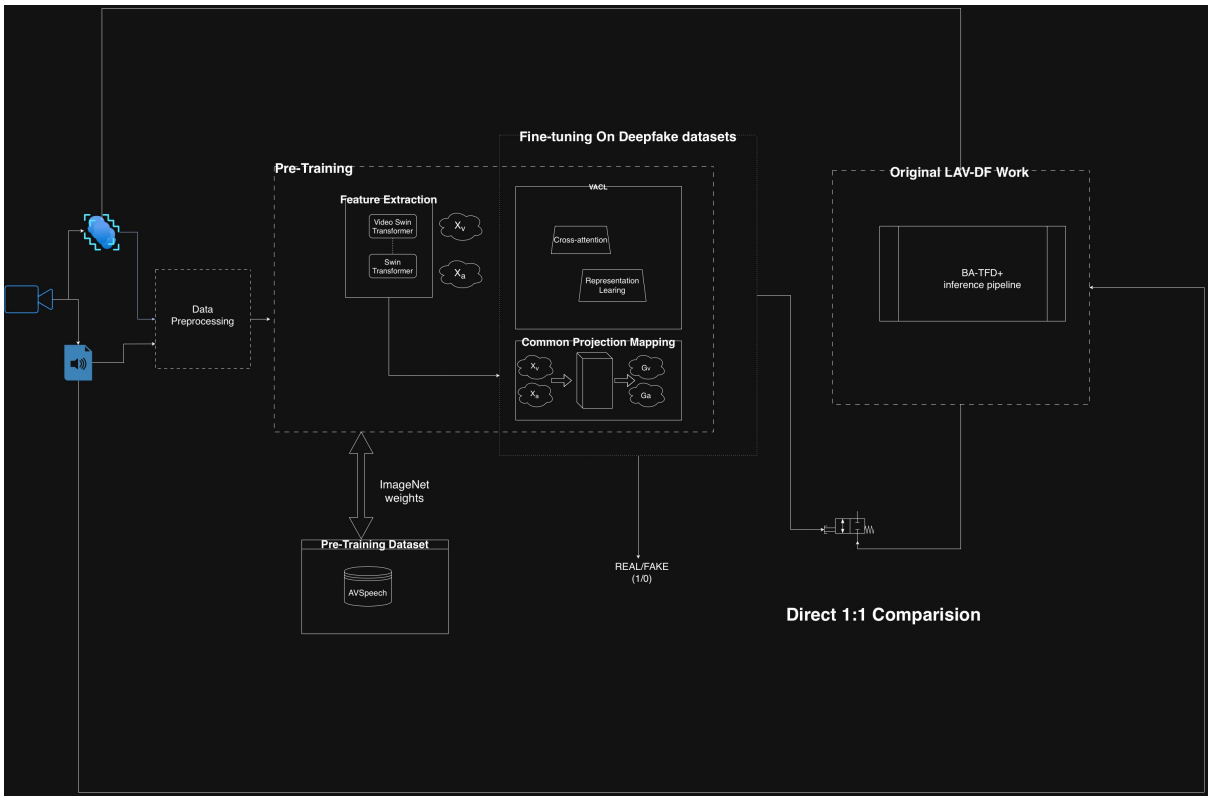


Figure 4: Model Architecture

The architecture comprises of the five main blocks. In the sequence -

1. Block 1 - **Feature Extraction**

   (a) **Video -** A Video Swin Transformer generates a tokenized batched representation $X_v \in R^{B \times T_v \times H' \times W' \times D_v}$ of arbitrarily picked video segments. The T corresponds to the *Temporal length*, H' & W' are the patched width and height of the video, and the D_v is the output dimension of the feature consisting of enriched features. This is the pre-pooling internal shape of the Video Swin Transformer. Th B is the size of the batch consisting of arbitrarily sampled video segments. The Video Swin Transformer was adopted to replace the original SWIN transformer in the paper. This is because the VST is better adapted at conducting learnt multi-head attention amongst a temporal sequence of frames than the Swin transformer which would require a significant adaptation.

   (b) **Audio -** A Vanilla-Swin Transformer generates a tokenized batched representation $X_a \in R^{B \times T_a \times D_a}$ of audio segments. This batched input is padded to a common T amongst the samples in the Batch. The Swin Transformer assumes image patches, and due to this the Audio is resized without interpolation or other rescaling techniques to safeguard enriched artefacts which may be lost in the process of feature learning to detect deepfake.

2. Module A - **Pre-Alignment tokenisation module**
   Manipulates the temporal-stage information in both the audio and video pairs by

reshaping them into a common plane of axis. The audio and video have to be pre-aligned before being fed into the VACL block. The alignment axis chosen is the sequence of time.

3. Module B (Branched from Block 1) **Visual-Audio Correlation Learning (VACL)** - This follows a direct paper-faithful implementation of the mathematics used in the original work.

4. Module C (Branched from Block 1) - **Common Projection Layer** This block projects the swin-adapted features into a common latent space to generate a common cross-modal plane containing both the modalities with an InfoNCE contrastive learning loss. The method aims to strengthen both modalities into a common learnt space which attends to both of them. This task is auxillary to the parallel branched VACL, as it trains with a contrastive level objective - so that they can be better adapted to be correlated meaningfully.

5. Module D - **Classification layer** This block takes the attention-based features from the Module B and passes through a Fully Connected layer to classify the video as real or fake. The loss function used is the BCE loss with a training weight of 1.0.

3.4.1 Model Training

The tables below show the divergences between original work and current work

Original Work	Current Work	Remarks
Uses a bi-phase training mechanism where 1. The architecture is first pre-trained on REAL videos 2. The architecture is then fine-tuned on the target group videos	Follows the same bi-phase contract in order to replicate the skeleton structure	-
Stage 1- The pre-training is conducted on the AVSpeech (Ephrat et al.; 2018) and the VoxCeleb2 (Chung et al.; 2018) dataset in a self-supervised learning mechanism.	The pre-training phase is only conducted on 15k videos of the AVSpeech	As the pre-training involved a non-labeled self-supervised learning contrastive-loss, which generates more energy in the presence of non-explicit labels.
Stage 2 - The fine-tuning strategy involve loading pre-trained weights and turning off the Swin-feature backbones. The swin-features are not learning as the model has been pretrained on millions of videos capturing more than 100 frames per video.	Use-case training experiments were conducted on both - although primarily the fine-tune stages involved turning off the backbone weights.	As both SWIN models were initialised from random-init the generalisation ability was tested by adapting the Stage2 model to train with and without backbone weights

Table 5: Model Training Overview

3.4.2 Energy Profiling

The carbon footprint of the self-supervised training module was monitored as a factor of the amount of energy consumed per device, the energy wattage of the training run, and the carbon emissions per training run. The metrics also log the peak memory consumption per GPU, with the goal of classifying the type of deep learning system in terms of Green AI goals.

4 Design Specification

The section shall explain end-to-end execution flows & give a low-level description of each module within one training or evaluation run. It will also construct the model training stages - with a brief description of their divergence.

4.1 Training Pipeline

The architecture was trained and evaluated in two stages for a majority of experiments -

- **Stage 1** - This is the pretraining phase. The NPV architecture was trained on REAL videos from the AVPSpeech dataset in order to learn early Visual-Audio correlations. This facilitates the mdoel to adapt to downstream deepfake detection tasks in a more efficient manner. This closely mirrors the training scheme of the original authors who performed downstream detection tasks by training on the AVSpeech as well as on another dataset Audio-Video dataset namely the Voxceleb2 (Chung et al.; 2018). While the original authors used 2.5 million videos, the current work made use of 12,785 curated videos capped at 175 frames (a maximum length of 7 seconds)
- **Stage 2** - The NPV checkpoint is then loaded into the new architecture variant. This is the Stage 1 architecture with an added BCE classification loss. This in turn, mirrors the training scheme used by the original work.

4.2 Data Preprocessing

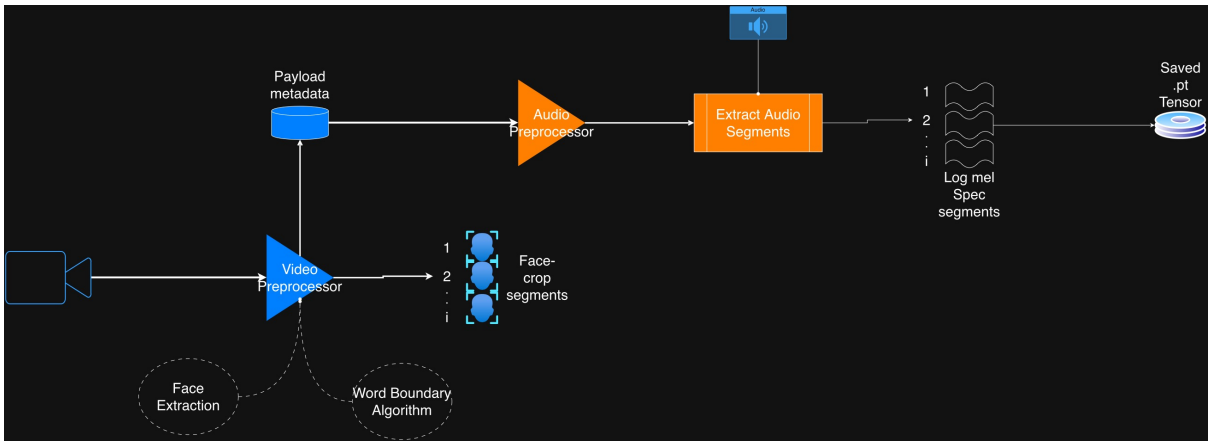


Figure 5: Data preprocessing flow outline

A word-boundary algorithm is used to generate word groups for spoken words in the video. The range of the lengths of each group is between [0.5-3 seconds]. This also directly corresponds to 75 frames as maximum length. The algorithm has been described below. The module uses an open-source Face Detector from the InsightFace library (Guo and Deng; 2025) to identify faces. The decoder script then generates word boundaries from the algorithm given below. This is used to mimic the phoneme-viseme setup in NPV Forensics.

$$Segment(i) = FaceDetection(WBG(i))$$

4.2.1 Word Boundary algorithm

The word boundary algorithm constructs aggregated segments based on time bounds. The words are grouped in order into "speech segments" by combining timestamps in the original video. The original authors use NPV segments, while the current work uses a greedy algorithm which ensures temporal coherence. The total length of each segment have to fall in between a minimum and maximum word length limit, which is calculated from the total frame length of the video. If the word group exceeds the length of the hard upper limit, the extra value is passed into the next segment and the current word group is added as the final segment. In the case the word group under-flows the hard limit, the current segment's extra length is appended to the next segment.

Algorithm 1 Pseudo-code for word boundary algorithm

Require: Timestamp range list $S = \{s^1, t^1], [s^2, t^2], \ldots, [s^N, t^S]\}$
Require: Video clip $V = \{v^1, v^2, \ldots, v^f\}$
$min\_bound \leftarrow 0.5$
$max\_bound \leftarrow 1.5$
min_len
for all $S[i] \in S$ **do**

  $segment\_begin \leftarrow FetchFirstWordOf(S[i])$
  $segment\_end \leftarrow FetchLastWordOf(S[i])$
  $segment\_next\_begin \leftarrow FetchNextFirstWordTo(S[i])$
  $segment\_next\_end \leftarrow FetchNextLastWordTo(S[i])$

  $word\_group\_len \leftarrow |segment\_next\_end| - |segment\_start|$
  if $|word\_group\_len| >= min\_len$ **then**
    if $|word\_group\_len| <= max\_bound$ **then**
      $segment\_start \leftarrow segment\_next\_start$
      $segment\_end \leftarrow segment\_next\_end$
      $AddToFinalList(S[segment\_start, segment\_end])$
    else
      . $AddToFinalList(S[segment\_start, segment\_end + max\_len])$
    end if
  end if
end for

4.3 Feature Extraction

4.3.1 Audio feature extraction

The saved .pt tensor payload is loaded from disk into the system by a dataloader optimised for Lightning setup. The features are log mel spectrogram with 96 time steps T and 64 mel bins. This contains pre-encoded temporal & frequency information.
As a crucial stage for the SWIN transformer to consume the audio without changing its architecture - the audio is simply converted into a 2-dimensional frequency-time representation. An extra dimension is added into the setup to represent the shape $1 \times 224 \times 224$. The process reshapes the spectrogram onto a pre-padded 224×224 canvas without resizing, rescaling or introducing artefacts by the means of interpolation. The shape then becomes - The final extracted feature shapes are given below -

$$x_a \in R^{B \times 1 \times 224 \times 224}$$

This is then passed into the feature extractor. The final extracted feature shapes are given below -

$$X_a \in R^{B \times L \times D_a}$$

4.4 Video feature extraction

Each swin-ready batch segment is obtained from decoding its corresponding video segment and normalisation. The video is fed into the vanilla swin transformer to obtained a pre-head intermediate feature token map. The shape is -

$$X_v \in R^{B \times D_v \times T_v \times H' \times W'}$$

4.5 Post-Swin pre-alignment of audio-video pairs

The batch-wise tokenized-sequence outputs of the Swin 2D and Video Swin Transformers are principally different to be passed into the VACL block. The 3-dimensional SWIN audio is concatenated into the 5D swin audio through non-learnable common projection & max pooling.

$$X_a \in R^{B \times L \times D_a}$$
$$X_v \in R^{B \times T' \times H' \times W' \times D_v}$$

textbfVideo spatio-temporal tokens $T' \times H' \times W'$ are summarised using global-average pooling into a single S sequence.

$$X_v(post - pooling) \in R^{B \times S \times D_v} = GlobalAvergePool(R^{B \times L \times D_a})$$

Audio frequency-temporal token L is reshaped to obtain it in a 2D matrix axis form of shape (2,2). This is then pooled an interpolated to match the video token sequence S.

$$X_a(post - pooling) \in R^{B \times S \times D_a} = Interpolation(Pool(Reshape(R^{B \times L \times D_a})))$$

4.5.1 VACL block & Common Projection Block)

Both modules are implemented as-is from the original work.

The VACL block generates final discriminative features in Stage 2.

4.6 Stage 1 model execution flow

For Stage 1 architecture - a paper faithful implementation is done on the self-supervised learning setup. The stage will oversee the model being trained on a few thousands of real videos to generalise with weights.

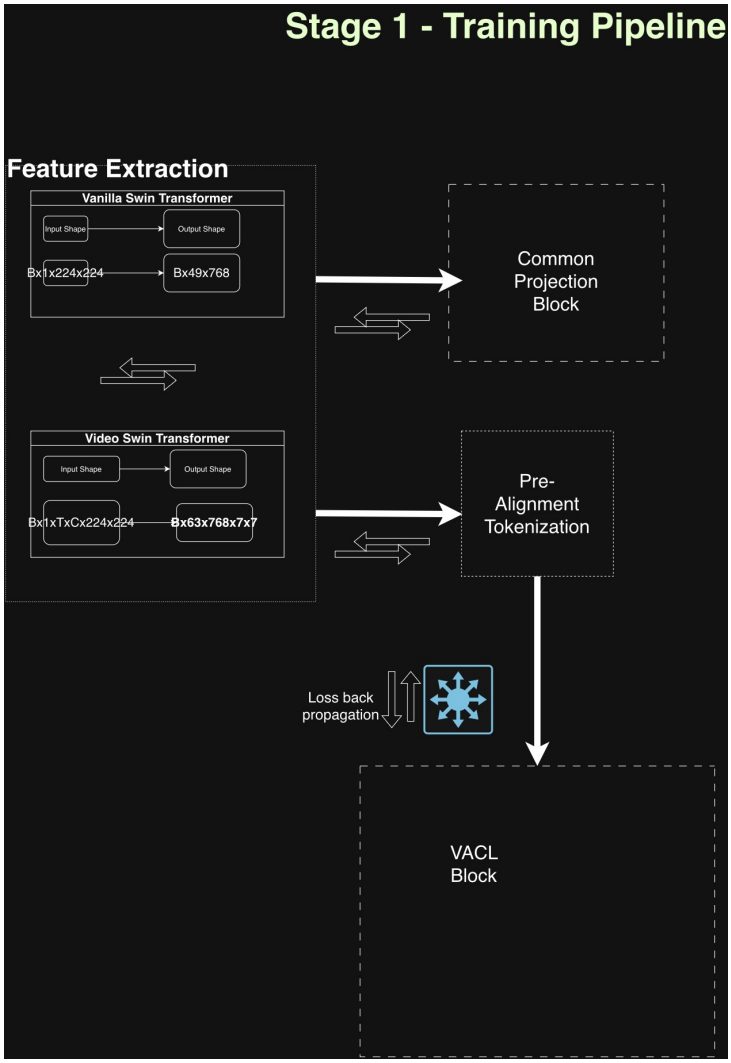


Figure 6: Stage 1 Current Work Pipeline

In this stage, the architecture is pretrained using Self-Supervised Learning, with the four loss functions.

$$\mathcal{L}_{total} = \mathcal{L}_{vacl} + \mathcal{L}_{cpe/infonce}$$

4.7 Stage 2 model execution flow

The stage 2 model is the final classification model. It generates the final logits for classification. The pretrained weights from the Stage 1 phase are loaded into the phase Stage 2. The backbone Swin weights are frozen to evaluate the model quality on learnt representation from the VACL block.

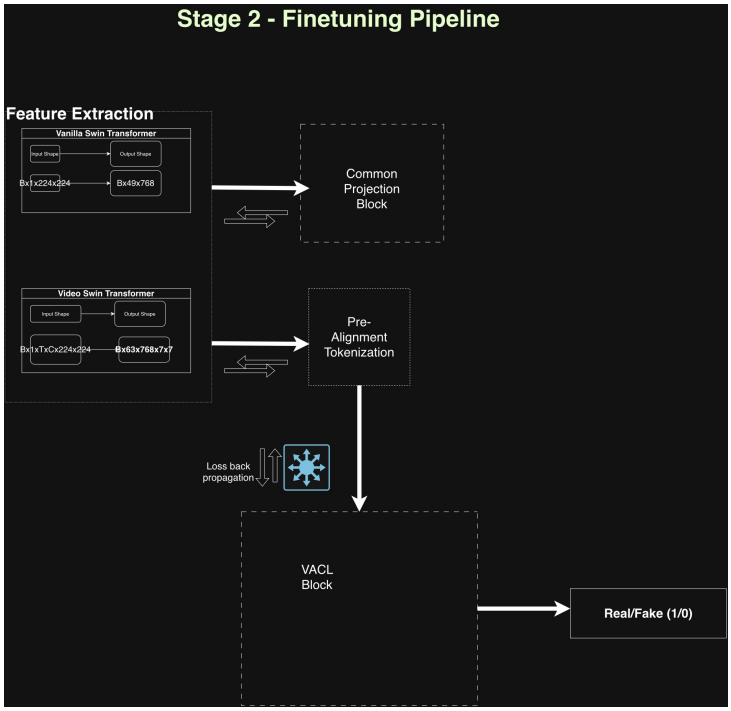


Figure 7: Stage 2 Fine-Tuning pipeline

The total loss function is observed as -

$$\mathcal{L}_{stage2-total} = \mathcal{L}_{vacl} + \mathcal{L}_{cpe/infonce} + \mathcal{L}_{cls}$$

4.8 Final Classification

Uses a MLP for final classification employing an Binary Cross Entropy loss function.

4.9 Dataset

The AVSpeech, DFDC & LAV-DF datasets are used in both the training & evaluation configurations.

- Stage 1 uses AVSpeech videos, and a transcribed word-timestamp file. The transcription is done using OpenAI's Whisper tool which requires a local installation.
- Stage 2 uses LAV-DF (Training set), DFDC (test set)& LAV-DF (testing set). A cross-combination of each of these have been used to indiviually train and validate the model in each of the training & evaluation schemes (described in later section).

  The REAL/FAKE distribution is set to 8:2 (80% are FAKE and 20% are REAL).

The total number of rows, Real/Fake distribution, lengths of fake segment(s) are depicted in the visuals below for both Stage 1 & 2 are showm below.

Table 6: Train/Val dataset group

Table 7: AVSpeech

Label	Segments
REAL(0)	12781
TOTAL	12781

Table 8: LAV-DF (Train)

Label	Segments
REAL(0)	7510
FAKE(1)	32328
TOTAL	39838

Table 9: Group C

Label	Segment
REAL(0)	587
FAKE(1)	3016
TOTAL	3603

Table 10: Group D

Label	Segments
REAL(0)	281
FAKE(1)	1193
TOTAL	1474

4.10 Model Training

The model training setup has been optimised to run on NVCC core GPU processors using Pytorch's Lightning API. The model is executed in 8 parallel branches - one for each shard distributed to GPU IDs [0,1....7] respectively.

5 Implementation

The models were trained on a A100 8x GPU setup using Lightning framework. This consists of 8 GPUs training in parallel. The multi-GPU setup was leveraged to save time and efficiency. For all experiments, tensorboard and csv logs were maintained. For each experiment run, the model checkpoint for the last epoch was saved in the tensorboard directory. There are a total of **11** training variant for Stage 2 & one training variant for Stage 1.
Additionally the experiments in Stage 2 are bracketed into either using frozen SWIN backbones while traning, using either of LAV-DF or DFDC to train, and whether the training runs utilise pre-trained stage 1 weights as an additional ablation study analysis. The training recipe for Stage 1 & 2 have been shown in different table groups below. There are two table sets - one for describing the training hyperparameters and the other the components overlay set used in the training runs.

5.1 STAGE 1 - Pre-training NPV model

A total of 12,981 videos and audio clips were used in training the model. Train/Loss curves were generated, along with an emissions file measuring electricity consumption and FLOPs for single GPU with id 0. The table depicts the hyper-parameters values used in training the self-supervised learning setup -

Hyperparameter	Value
Learning Rate (lr)	1e-4
Model weight decay	1e-2
Batch size	4
Epochs	5

Table 11: Stage 1 pretraining — hyperparameter values

5.2 STAGE 2 - Fine-Tuning Stage 1 model using various combinations of dataset + pretrained weight usage

The experiments load the Stage 1 graph from the tensorboard log checkpoints. The model of the last epoch and final step are loaded. The experiment scope is further divided into either training with frozen SWIN backbone, or training without it. The ablation experiments divide between pretrained Stage 1 usage vs training Stage 2 without stage 1 pretrained weights. These are outlined in the component overlay table-set below

Hyperparameter	Value
Learning Rate (lr)	5e-4
Model weight decay	2e-4
BCE function weight	1.0
VACL block weight	0.9
CPE block weight	0.4
Batch size	4
Epochs	5
Binary classifier Neural Network (number of neurons)	256

Table 12: Stage 2 Configuration A - Frozen Swin Backbones

Hyperparameter	Value
Learning Rate (lr)	1e-6
Model weight decay	1e-2
BCE function weight	0.6
VACL block weight	0.8
CPE block weight	0.3
Batch size	4
Epochs	5
Binary classifier Neural Network (number of neurons)	128

Table 13: Stage 2 Configuration B - Trainable Swin Backbones

The table groups have been summarized in the Training set. Each of them train on either the LAV-DF Training set or the DFDC test set. The pretrained weights from Stage 1, as described earlier, may or may not be loaded into the Stage 2 training paradigm.

Component	Value
Training Set	LAV-DF
Pretrained Stage1	Yes
Frozen SWIN	Yes

Table 14: Stage 2 Finetuning — Config A

Component	Value
Training Set	LAV-DF
Pretrained Stage1	Yes
Frozen SWIN	No

Table 15: Stage 2 finetuning — Config B

Component	Value
Training Set	LAV-DF
Pretrained Stage1	No
Frozen SWIN	Yes

Table 16: Stage 2 Finetuning — Config C

Component	Value
Training Set	LAV-DF
Pretrained Stage1	No
Frozen SWIN	No

Table 17: Stage 2 finetuning — Set D

Component	Value
Training Set	DFDC
Pretrained Stage1	Yes
Frozen SWIN	No

Table 18: Stage 2 pretraining — Set E

Component	Value
Training Set	DFDC
Pretrained Stage1	No
Frozen SWIN	Yes

Table 19: Stage 2 pretraining — Set F

Component	Value
Training Set	DFDC
Pretrained Stage1	Yes
Frozen SWIN	Yes

Table 20: Stage 2 pretraining — Set G

Component	Value
Training Set	DFDC
Pretrained Stage1	No
Frozen SWIN	No

Table 21: Stage 2 pretraining — Set H

6 Evaluation

The evaluation scheme for the NPV-modified model uses every configuration from the training scheme to evaluate on non-identical dataset(s). This has been summarised in the table below.

Training Configuration	sDataset
A	LAV-DF - Testing set
D	LAV-DF - Testing set
G	DFDC
H	DFDC

Table 22: NPV Stage 3 - Evaluation Rule-set

6.1 Evaluating the original BA-TFD+ model

The BA-TFD+ pretrained model is evaluted using the metrics described below. The dataloader applies the same sampling strategy for the BA-TFD+ model to be able to generate a direct point of evaluation of NPV-modified model.

6.2 Green Energy Evaluation

Graphs were generated for the Stage 1 run through GPU monitoring & carbon footprints logged via code carbon and GPU memory tracking for the entire run. A time-series graph showing memory usage across the run was logged for a single GPU id [0]. This was correlated with the code carbon logs to validate the model for energy efficiency.

6.3 Evaluation Metrics

Every configuration case is evaluated by the metrics below. The following common metrics will be used. They are -

1. Precision-Recall curve - Following from the above correspondence a measure of how many classifications of Fake videos are actually Fake.

2. AUCROC curve - The Recall is the measurement of how many Fake videos which are classifed as are in actuality fake.

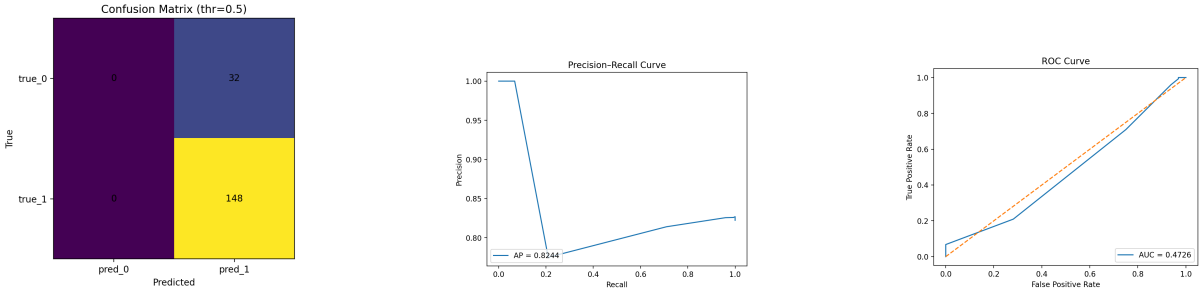
3. Confusion Matrix - It gives the exact make-up of how many Truly classified Fakes are Fakes (True Positives), how many falsely classified FAkes are actually fakes (False Positives), and subsequently how many Truly classified REAL videos are REAL (True Negatives), how many falsely classified REAL videos are actually REAL (False Negatives).

7 Results

7.1 Qualitative Evaluation

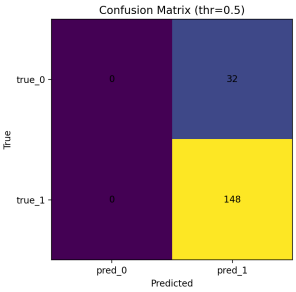
7.1.1 NPVForensics Model

The PR, ROC-AUC and the CM have been published for configurations A, B, C, D, E& F.

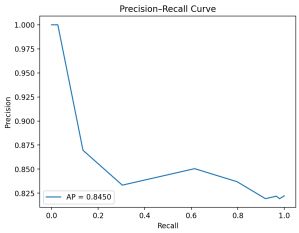


(a) Confusion Matrix (b) PR Curve (c) ROC Curve

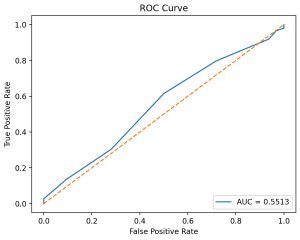
Figure 8: Config A results



(a) Confusion Matrix

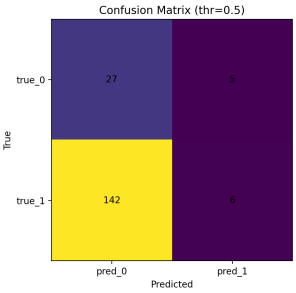


(b) PR Curve

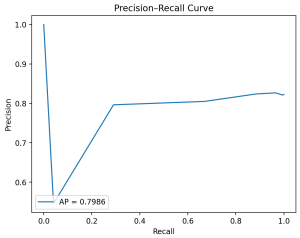


(c) ROC Curve

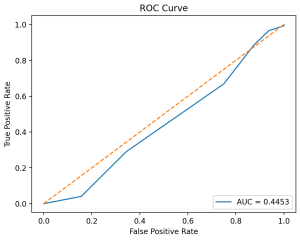
Figure 9: Config D results



(a) Confusion Matrix

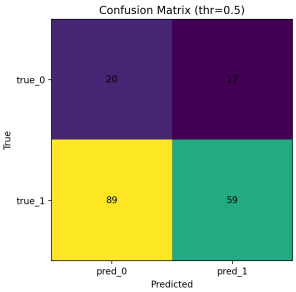


(b) PR Curve

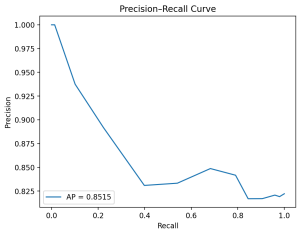


(c) ROC Curve

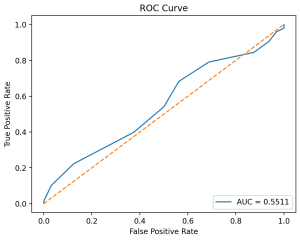
Figure 10: Config G results



(a) Confusion Matrix



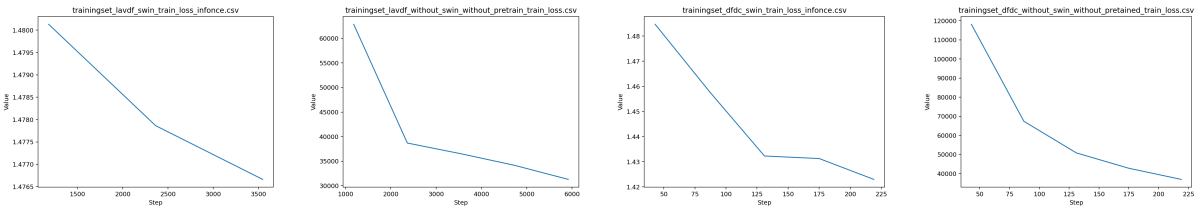
(b) PR Curve



(c) ROC Curve

Figure 11: Config H results

The loss curves are displayed for the key experiments -



(a) A1 Train (b) D Train (c) G Train (d) H Train

Figure 12: Training loss curves (1×4) for Configurations A1, D, G, and H.

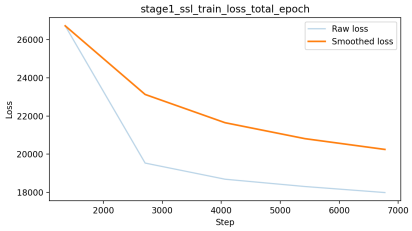


Figure 14: Stage 1 Training loss

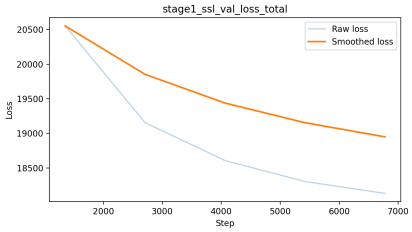
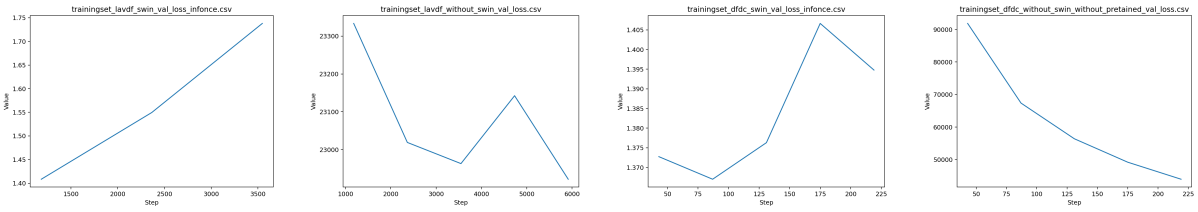


Figure 15: Stage 1 Validation Loss



(a) A1 Val (b) D Val (c) G Val (d) H Val

Figure 13: Validation loss curves (1×4) for Configurations A1, D, G, and H.

7.1.2 Energy efficiency

There are two groups of visuals for emissions and energy consumption. The pie charts describe the ef

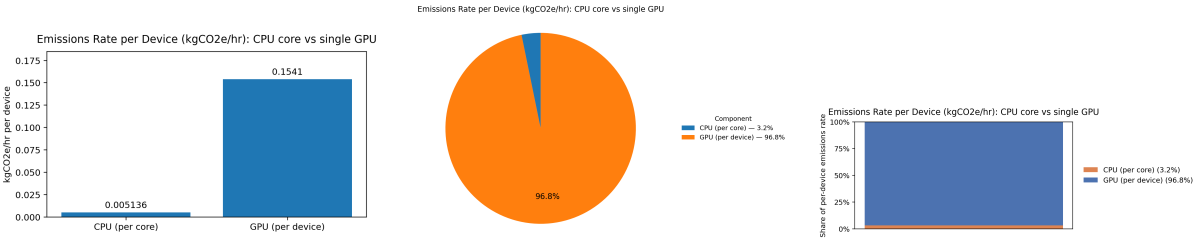


Figure 16: Emissions: GPU/CPU use-scenarios

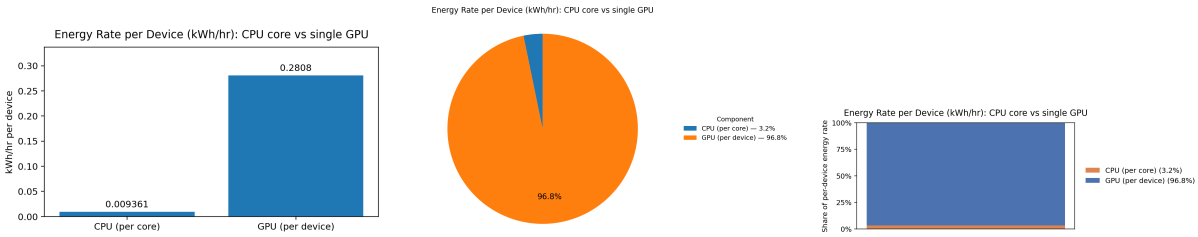
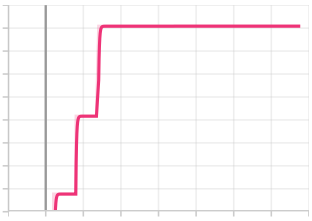
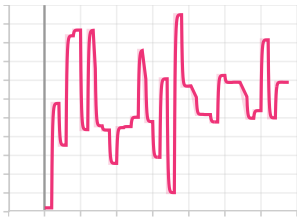


Figure 17: Energy: GPU/CPU use-case scenarios



(a) Memory Reserved



(b) Memory Allocated

The visuals depict different graphs sets for energy & emissions utilisation. The energy/ emissions rate per single GPU/CPU cores per single hour, their distributions and contribution towards both energy & emissions per device, per hour and in general. There are also graphs for 18b memory allocated & memory held reserved 18a to a single GPU during the Stage 1 training run. It was observed -

- The GPU utlised an equivalent amount of share, a vast majority 96.8% against a single CPU core. It also consumed the same share of energy.
- It also contributed an equivalent distribution of energy/emissions per single core/device.
- Absolute values show the massive differences between energy/emissons emittance, which is expected.

Further, the loss/training graphs shown for stage 1 in 15, 14 depict the architecture as partially learning. The 18b curve shows the memory is allocated haphazardly during the training run spanning 5 epochs, although not constantly. Figure 18a indicates a stable memory convergance, after a few 1000 steps .

We can further infer from the second diagram, the 96.8% core-share, and given the pipeline is offline training this means the GPU contributed to the model work - and the emissions and consumptions 96.8% of the time

It can thus be inferred the setup scaled to 8 GPUs is a reasonable trade-off between consumption against performance with the objective of the model to commit to partial learning. It is green-AI positive as the environmental-efficient GPU cores although consume energy 96.8% of the time, the decrease in the train/val loss with stable plateau-ing of the memory reserved graph indicate an effective usage of the GPU memory space. From the graphs we can infer the models are not training or training sub-optimally.

- The AUC curves exhibit average or above average performance. Most of them show under or equal to 0.50 which is marginally worse or equal to random guessing. With the exceptions of cases in Case H 11.

- THe confusion matrices show a degenerate rule. Most of them predict or classify all segments into a single label. With the exception of fig Case H 11, which shows balanced classes albeit with a low accuracy - (the case where a non-pretrained Stage 1 is trained on DFDC test set with a trainable/frozen Swin backbone and evaluation is performed on a test set of LAV-DF)

- The Precision recall sensitivity curve - shows a high to above high performance on all the cases despite the matrices and AUC showing a low performance. This shows a higher confidence on positive prediction (FAKE labels), for each model case. This trend remains consistent even for the LAV-DF cases highlighted earlier 8, 9.

The train/loss curves are shown for different training configs, out of which the Config H with non-learnt DFDC case shows initial signs of learning, as both the train and val losses drop consistently.

8 Conclusion and Future Work

The end goal of this work was to assess a SOTA model on a temporal localised dataset in a resource-constrained environment. The elimination of the LFA block, the additional "lip" stream and the generated graphs ontwards classify the architecture as a good trade off between performance & sustainability.

Most models did not seem to learn optimally. The extreme shifts in the PR curves, high fake retrieval confidence as well on all models indicates the models assigned a high confidence to detect the fake samples. However the PR curve does not guarantee learning and should be cross-referenced with the AUROC curve. The representation were sub-par, as extrapolated from the AUCROC curve for all LAV-DF trained cases. Moreover the transfer of weights learnt by the AVSpeech initial training phase did not yield representational space suited to the task of LAV-DF.
The best performing model, however was the Config G (without swin and no pretrain weight transfer; trained on DFDC test set). Evaluation on the LAV-DF test set confirms the models generalizable ability on the new localised dataset. This is also attested by the train/loss graphs indicate partial model-learning. The spread across all quadrants indicate a positive starting point, as well as ¿0.5 AUCROC score, and a high PR sensitivity graph.

However, at the same time the pretrained BA-TFD+ model produced a low ROC curve. This can be justified with the low segment length (75 frames) of the video segments, whereas the original architecture was trained on 512 frames. The inferred boundary maps as a result did not yield positive discriminations. This signifies the future problem to be solved as a more novel "sub-type" of temporal localisation happen in compact video length space. In conclusion, The NPV model was built to analyse NPV segments which relied on cues across segments. However as the boundary-focused LAV-DF dataset does not rely on contextual continuity, the ordering was not given a priority. As a result, the NPV model failed to learn discriminations in the out-of-domain LAV-DF space. As future work, the model could be re-trained on a large dataset, along with more cross-dataset use-cases.

References

Alrashoud, M. (2025). Deepfake video detection methods, approaches, and challenges, *Alexandria Engineering Journal* **125**: 265–277.

Brian Dolhansky, Joanna Bitton, B. P. (2020). The DeepFake detection challenge dataset.

Cai, Z., Ghosh, S., Dhall, A., Gedeon, T., Stefanov, K. and Hayat, M. (2023a). Glitch in the matrix: A large scale benchmark for content driven audio–visual forgery detection and localization, *Computer Vision and Image Understanding* **236**: 103818.

Cai, Z., Ghosh, S., Dhall, A., Gedeon, T., Stefanov, K. and Hayat, M. (2023b). Glitch in the Matrix: A Large Scale Benchmark for Content Driven Audio-Visual Forgery Detection and Localization.

Chen, Y., Yu, Y., Ni, R., Li, H., Wang, W. and Zhao, Y. (2025). NPVForensics: Learning VA correlations in non-critical phoneme–viseme regions for deepfake detection, *Image and Vision Computing* **156**: 105461.

Chung, J. S., Nagrani, A. and Zisserman, A. (2018). VoxCeleb2: Deep Speaker Recognition, *Interspeech 2018*, pp. 1086–1090.

Dolhansky, B., Bitton, J., Pflaum, B., Lu, J., Howes, R., Wang, M. and Ferrer, C. C. (2020). The DeepFake Detection Challenge (DFDC) Dataset.

Dosovitskiy, A. (2020). An image is worth 16x16 words: Transformers for image recognition at scale, *arXiv preprint arXiv:2010.11929* .

Ephrat, A., Mosseri, I., Lang, O., Dekel, T., Wilson, K., Hassidim, A., Freeman, W. T. and Rubinstein, M. (2018). Looking to Listen at the Cocktail Party: A Speaker-Independent Audio-Visual Model for Speech Separation, *ACM Transactions on Graphics* **37**(4): 1–11.

Gui, J., Chen, T., Zhang, J., Cao, Q., Sun, Z., Luo, H. and Tao, D. (2024). A Survey on Self-Supervised Learning: Algorithms, Applications, and Future Trends, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **46**(12): 9052–9071.

Guo, J. and Deng, J. (2025). Deepinsight/insightface, Deep Insight.

Habgood-Coote, J. (2023). Deepfakes and the epistemic apocalypse, *Synthese* **201**(3): 103.

Ilyas, H., Javed, A. and Malik, K. M. (2023). AVFakeNet: A unified end-to-end Dense Swin Transformer deep learning model for audio–visual deepfakes detection, *Applied Soft Computing* **136**: 110124.

Lewis, J. K., Toubal, I. E., Chen, H., Sandesera, V., Lomnitz, M., Hampel-Arias, Z., Prasad, C. and Palaniappan, K. (2020). Deepfake Video Detection Based on Spatial, Spectral, and Temporal Inconsistencies Using Multimodal Deep Learning, *2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pp. 1–9.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows, *arXiv preprint arXiv:2103.14030* .

Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S. and Hu, H. (2021a). Video Swin Transformer.

Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S. and Hu, H. (2021b). Video swin transformer, *arXiv preprint arXiv:2106.13230* .

Su, H., Gan, W., Wu, W., Qiao, Y. and Yan, J. (2021). BSN++: Complementary Boundary Regressor with Scale-Balanced Relation Modeling for Temporal Action Proposal Generation.

Tmamna, J., Ayed, E. B., Fourati, R., Gogate, M., Arslan, T., Hussain, A. and Ayed, M. B. (2024). Pruning Deep Neural Networks for Green Energy-Efficient Models: A Survey, *Cognitive Computation* **16**(6): 2931–2952.

Wang, M. (2025). Deepfake Detection: A Multimodal Survey, *ITM Web of Conferences* **78**: 02027.

Weerawardana, MC. and Fernando, TGI. (2021). Deepfakes detection methods: A literature survey, *2021 10th International Conference on Information and Automation for Sustainability (ICIAfS)*, pp. 76–81.