Audio Transformer for Synthetic Speech Detection via Multi-Formant Analysis

Luca Cuccovillo, Milica Gerhardt, and Patrick Aichroth
Fraunhofer Institute for Digital Media Technology IDMT
Ehrenbergstraße 31, 98693 Ilemanu, Germany
{luca.cuccovillo, milica gerhardt, patrick.aichroth}@idmt.fraunhofer.de

Abstract

This paper introduces a novel multi-task transformer for detecting synthetic speech. The network encodes magnitude and phase of the input speech with a feature bottleneck, used to autoencode the input magnitude, to predict the trajectory of the first phonetic formants (F0, F1, F2), and to distinguish whether the input speech is synthetic or natural. The approach achieves state-of-the-art performance on the ASVspoof 2019 LA dataset with an AUC score of 0.932, while ensuring interpretability at the same time.

1. Introduction

Over recent years, advances in synthetic speech generation have reached remarkable milestones, to the point that it is now possible to reproduce highly realistic voices which are hard to detect even for trained professionals [17]. Hence, widespread access to advanced speech synthesis models and tools to the larger public has sparked considerable concerns over potential misuse of such technology for creating disinformation [2], and for its potential of becoming a staple tool for organised crime [9]. These worries are exacerbated by the current lack of effective and robust detection methods, compounded by limited availability of suitable training content, insufficient cooperation between industry and academia, and significant challenges with respect to ensuring generalizability of detection approaches with respect to unknown synthesis algorithms [4].

Artificial intelligence (AI) has played a key role in developing state-of-the-art detectors, and significant work was conducted within the community associated with Automatic Speaker Verification and Spoofing Countermeasures Challenge (ASVspoof): Models crafted by the challenge organizers, such as RawNet2 [25], RawGAT-ST [26], AASIST [13], have laid a solid foundation for synthetic speech detection.

However, given the data-dependant nature of these models, interpretability is a challenge: It is not possible to determine ex-ante which input features will end up being relevant for the detection. Salvi *et al*. [21] aimed at a post-hoc analysis, using methods for explainable AI devised for image input processing [20, 24]. They discovered that the networks were primarily influenced by non-vocal spectrogram regions – silent parts, and very low and very high frequency ranges. This finding was validated by a subsequent study from the same research team, which achieved state-of-the-art synthetic "speech" detection performance by focusing on the analysis of background noise alone, without considering the speech content [23].

The ability to explain the decision-making process of utilized networks is not only a nice-to-have feature but a mandatory one, especially when it comes to forensics examination for the court. This is required, for instance, by the upcoming European Artificial Intelligence Act [3] which classifies all techniques that could impact the citizens' rights and freedom, including tools used for evidence analysis in legal trials, as high-risk AI. Hence, there is a pressing need for detection systems *designed* to be explainable, and to focus on the modeling and analysis of speech signals and their characteristics.

We described and tried to satisfy these requirements in [5] by proposing SFAT-Net, i.e., an audio transformer for speech formant analysis based on the hypothesis that the energy distribution among vocal formants of synthetic speech exhibits anomalies. In the initial paper, we designed a multi-task architecture, tailored for audio signals, that included a feature bottleneck. This bottleneck was utilized to autoencode the input spectrogram, to predict the fundamental frequency (F0) trajectory of the input utterance, and then to classify the input speech as synthetic or natural.

SFAT-Net relied on sequence-to-sequence (seq2seq) transformers as basic building block, thereby benefiting from their attention mechanism [28]. Indeed, attention is well suited for predicting F0 and autoencoding the input, thanks to the direct correlation between input speech harmonics and fundamental frequency, which is essential for the correct reconstruction of the input. Thus, speech synthesis detection in SFAT-Net can be considered a byproduct of attention to the energy distribution among vocal for-

mants, and the approach provides a clear understanding about which characteristics of the input signal contribute to the final outcome.

The encoding part of the model, initially based on a visual transformer applied to the audio spectrogram [7], was later extended in SFAT-Net-2 [6] to include an additional phase input: By employing a shared framing grid and positional embeddings, the modified network achieved better performance than the baseline despite the lower number of parameters, maintaining the feature bottleneck unchanged. Building on this achievement, we now propose to extend the SFAT-Net concept by revising the decoding process.

Even though F0 is a fundamental characteristic of the input speech, we believe that a more appropriate approach should also encompass other characteristics of the input utterances, and in particular the F1 and F2 formants that in phonetic analysis are relied upon to describe the speech content [12], the speaker identities [18], and the speaker voice qualities [15]. Thus, In this paper we introduce the SFAT-Net-3 architecture, which enhances the previous versions by adopting a more sophisticated decoder capable of reconstructing not only the F0 trajectory but also the trajectories of the F1 and F2 formants, thereby offering a deeper analysis of the input speech.

The rest of the paper is organized as follows: Section 2 provides a comprehensive overview over the model, introducing the new decoder and its corresponding loss function. Section 3 outlines the evaluation setup used in our experiments, the results of which are presented in Section 4. Finally, Section 5 concludes the paper with a summary of our findings, and directions for future research.

2. Proposed Architecture

Our architecture consists of five key components, outlined below and illustrated in Figure 1:

1. Magnitude Encoder E_{X}: A seq2seq transformer converting the magnitude X of an input file into a suitable sequence of embeddings $y^{\mathrm{X}}_{\mathrm{enc}}$ (Section 2.1).
2. Phase Encoder E_{Φ}: A seq2seq transformer converting the phase Φ of an input file into a suitable sequence of embeddings y^{Φ}_{enc} (Section 2.2).
3. Magnitude Decoder D_{X}: A seq2seq transformer converting the joint encoding embeddings y_{enc} in an approximation of the input log-magnitude (Section 2.3).
4. Multi-formant Decoder D_F: A seq2seq transformer converting y_{enc} in an approximation of the trajectories of the F0,F1,F2 formants of the input speech (Section 2.4).
5. Synthesis Predictor P: A seq2seq transformer converting y_{enc} into a 2-dimensional vector indicating the presence of synthetic speech (Section 2.5).

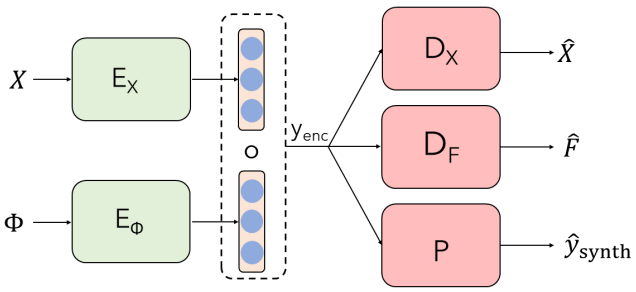


Figure 1. Proposed SFAT-Net-3 architecture

2.1. Magnitude Encoder

The magnitude encoder E_{X} is a seq2seq transformer designed to convert an input audio signal into a corresponding sequence of log-magnitude embeddings denoted as $y^{\mathrm{X}}_{\mathrm{enc}}$.

The input to the encoder is the log-spectral magnitude X obtained by the input recording x, i.e.:

$$X \in \mathbb{R}^{L\times M} = \log\left(\,|\mathrm{STFT}(x)|\,\right), \tag{1}$$

with L representing the number of frames and M the number of frequency bins of the Short-Time Fourier Transform (STFT).

The log-magnitude X is first split into non-overlapping 2D patches x^{X}_p, which are then projected into a series of patch embeddings $\mathbf{z}^{\mathrm{X}}$:

$$\mathbf{z}^{\mathrm{X}} \in \mathbb{R}^{N\times D^{\mathrm{X}}} = [z^{\mathrm{X}}_1, z^{\mathrm{X}}_2, \ldots, z^{\mathrm{X}}_N], \tag{2}$$

where

$$z^{\mathrm{X}}_p \in \mathbb{R}^{1\times D^{\mathrm{X}}} = \mathrm{project}(x^{\mathrm{X}}_p, \Theta^{\mathrm{X}}_{\mathrm{enc}}), \tag{3}$$

with $\Theta^{\mathrm{X}}_{\mathrm{enc}}$ being the required set of encoding parameters.

The log-magnitude embeddings $y^{\mathrm{X}}_{\mathrm{enc}}$ are calculated by feeding the sequence with positional information into an encoding transformer $T^{\mathrm{X}}_{\mathrm{enc}}$:

$$y^{\mathrm{X}}_{\mathrm{enc}} \in \mathbb{R}^{N\times D^{\mathrm{X}}} = T^{\mathrm{X}}_{\mathrm{enc}}(\mathbf{z}^{\mathrm{X}} + \mathbf{z}_{\mathrm{pos}}), \tag{4}$$

with $\mathbf{z}_{\mathrm{pos}}$ being standard learnable 1D positional embeddings described in the transformer architecture [28].

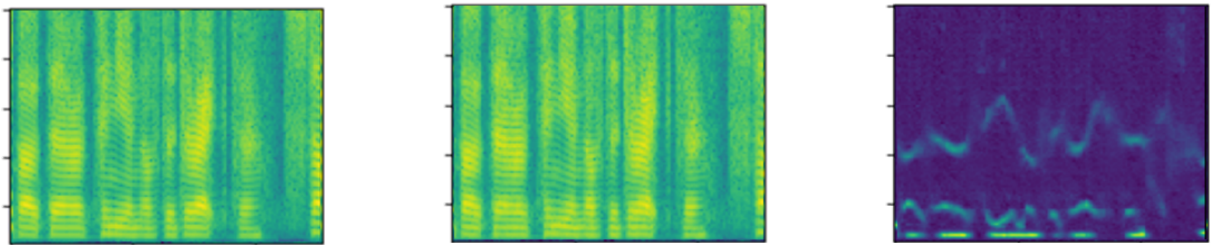
The transformer $T^{\mathrm{X}}_{\mathrm{enc}}$ consists of alternating layers of multi-headed self-attention (MSA) and multilayer perceptron (MLP) blocks, with layer normalization applied before, and residual connections after each block. Further details are provided in the related SFAT-Net-2 paper [6].

2.2. Phase Encoder

The phase encoder E_{Φ} is a seq2seq transformer designed to convert an input audio signal into a corresponding sequence of phase embeddings, denoted by y^{Φ}_{enc}.

The input to the encoder is the phase Φ obtained by the input recording x, i.e.:

$$\Phi \in \mathbb{R}^{L\times M} = \sin\left(\,\angle\,\mathrm{STFT}(x)\,\right), \tag{5}$$



(a) Input log-magnitude X (b) Reconstructed log-magnitude $\hat{X}$ (c) Estimated multi-formant matrix $\hat{F}$

Figure 2. Example outputs of the SFAT-Net-3 decoders, alongside the input log magnitude

where $\angle$ represents the phase of the STFT, and the sine function ensures that the signal is bounded.

Following a similar approach as for the magnitude embeddings, the phase Φ of the input STFT is initially split into non-overlapping 2D patches x_p^Φ, which are then projected into a series of patch embeddings $\mathbf{z}^\Phi$:

$$\mathbf{z}^\Phi \in \mathbb{R}^{N \times D^\Phi} = [z_1^\Phi, z_2^\Phi, \ldots, z_N^\Phi], \tag{6}$$

where

$$z_p^\Phi \in \mathbb{R}^{1 \times D^\Phi} = \text{project}(x_p^\text{X}, \Theta_\text{enc}^\Phi), \tag{7}$$

with Θ_enc^Φ being the required set of phase encoding parameters.

The phase embeddings y_enc^Φ are calculated by feeding the sequence with positional information into an encoding transformer T_enc^Φ:

$$y_\text{enc}^\Phi \in \mathbb{R}^{N \times D^\Phi} = T_\text{enc}^\Phi(\mathbf{z}^\Phi + \mathbf{z}_\text{pos}), \tag{8}$$

with the transformer T_enc^Φ being again a series of alternating layers of MSA and MLP blocks where layer normalization is applied before and residual connections after each block. Moreover, $\mathbf{z}_\text{pos}$ are the identical positional embeddings used by the log-magnitude encoder E_X – implying that the spatial information is shared across the two branches. Further details are provided in the SFAT-Net-2 paper [6].

2.3. Spectrogram Decoder

The spectrogram decoder D_X aims at mapping a sequence of embeddings $y_\text{enc} \in \mathbb{R}^{N \times D}$ to a matrix $\hat{X} \in \mathbb{R}^{LXM}$, which should closely resemble the log-magnitude X of the input recording.

In this paper, we define the sequence of encoding embeddings y_enc as

$$y_\text{enc} \in \mathbb{R}^{N \times D} = \left(y_\text{enc}^\text{X} \circ y_\text{enc}^\Phi\right), \tag{9}$$

meaning that the embeddings are created by concatenating the outputs of the encoders E_X and E_Φ alongside the last dimension, to obtain a sequence of embeddings in which each patch of the input is described by a $1 \times D$ vector.

These embeddings are processed by a decoding transformer T^X to yield a reconstructed sequence of log-magnitude patch embeddings:

$$\hat{\mathbf{z}}^X = T^X(\mathbf{z}^X + \mathbf{z}_\text{pos}^X), \tag{10}$$

which are then projected back into a time-frequency domain:

$$\hat{X} = \text{project}^{-1}(\hat{\mathbf{z}}^X, \Theta_\text{dec}^\text{X}), \tag{11}$$

where $\Theta_\text{dec}^\text{X}$ represents the necessary parameters. Further details are provided in the SFAT-Net-2 paper [5].

To ensure that the decoded output $\hat{X}$ matches the input log-magnitude X as closely as possible, i.e.,

$$D_\text{X}(y_\text{enc}, \Theta_\text{dec}^\text{X}) = \hat{X} \approx X, \tag{12}$$

the component must minimize an autoencoding loss during training:

$$l_\text{auto}\left(X \mid \Theta_\text{enc}, \Theta_\text{dec}^\text{X}\right) = \left\| X - D_\text{X}\Big(E(X, \Theta_\text{enc}), \Theta_\text{dec}^\text{X}\Big)\right\|_2, \tag{13}$$

with $\|\cdot\|_2$ denoting the l^2 norm, and Θ_enc denoting the union of the parameters of both encoders. An illustrative output from this process, alongside the corresponding log-magnitude of the input speech, is shown in Figure 2.

2.4. Multi-formant Decoder

In a nutshell, speech formants are resonant frequencies present in the acoustic signal of human speech, corresponding to specific resonance frequencies of the vocal tract. They are created by the unique shape and configuration of the vocal tract, including the throat, mouth, and nasal passages, as air passes through them during speech production.

Figure 3 shows F0, F1, and F2, known as the lower formants, as detected by the Praat software for phonetic analysis of speech [1]. These formants play a crucial role in speech perception, affecting the quality of vowels and certain consonants: Since they are influenced by the position and shape of the articulators (such as the tongue, lips, and jaw) during speech production, different speech sounds have characteristic formant patterns that help distinguish one sound from another.

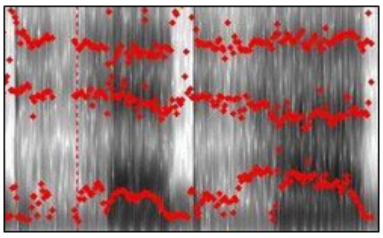


Figure 3. F0, F1, F2 formants detected by the Praat software [1]

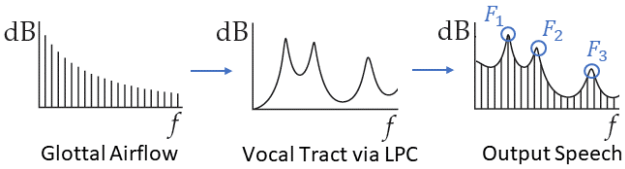


Figure 4. Formants retrieval via LPC analysis-by-synthesis

In the following, we describe the multi-formant decoder D_{F}, i.e., a component meant to map a sequence of embeddings $y_{\text{enc}} \in \mathbb{R}^{N \times D}$ to a matrix $\hat{F} \in \mathbb{R}^{LXM}$, estimating the trajectories of the F0, F1, F2 formants of the input speech.

Let us denote with $f_0(l)$, $f_1(l)$ and $f_2(l)$ the frequency values of the F0, F1 and F2 formants at a specific time frame l. All the information about these three components is described with a respective contour matrix $F_n \in \mathbb{R}^{L \times M}$ by means of:

$$F_n(l,m) = \begin{cases} 1 & \text{if } f_n(l) \approx \frac{f_s}{2M} \cdot m \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

with $l \in [1, L]$. Therefore, the task of the multi-formant decoder can be considered a regression problem, in which y_{enc} must be mapped to a superposition of the contour matrices of the formants of interest, i.e.:

$$D_{\text{F}}(y_{\text{enc}}, \Theta^{\text{F}}_{\text{dec}}) = \hat{F} \approx (F_0 + F_1 + F_2), \tag{15}$$

with $\Theta^{\text{F}}_{\text{dec}}$ the set of model parameters.

A similar task, albeit only tracking the contour for F0, was performed also by SFAT-Net. Therefore, we relied on the same architecture used for the spectrogram reconstruction to train a different transformer with its own projection matrices and learnt positional embeddings. Similarly, the loss of the network was based on the desired multi-formant contour matrix:

$$l_F\left(X, F \mid \Theta_{\text{enc}}, \Theta^{\text{F}}_{\text{dec}}\right) = \left\| F - D_{\text{F}}\Big(E(X, \Theta_{\text{enc}}), \Theta^{\text{F}}_{\text{dec}}\Big) \right\|_2, \tag{16}$$

with $\|\cdot\|_2$ denoting the l^2 norm.

In order for this task to be successful, the creation of a proper ground truth for the three formants of interest is crucial. Hence, we decided to rely on the formant estimation procedure provided by the Praat software for phonetic analysis [1], which makes use of the analysis-by-synthesis paradigm. In a first step, it estimates the linear predictive coding (LPC) coefficients of the input, by means of the Burg algorithm for auto-regressive modeling of time series [10]. It then re-synthesizes the input speech using the detected LPC coefficients, and identifies the location of the spectral peaks present for each output frame. The n-th detected formant corresponds to the n-th peak in the spectrum. A schematic visualization of the formants interpreted as peaks of the LPC-estimated speech is depicted in Figure 4.

Furthermore, following the example of the initial SFAT-Net, we decided to input the loss function calculation with a smoothed version of the multi-formant contour matrix F, rather than the formulation in Eq. (14). Therefore, we *filtered* F with a 3×3 Gaussian kernel with unitary variance, with the goal of penalizing small errors less severely than large ones, and of improving the convergence speed. An example of an estimated multi-formant contour matrix F obtained following the entire procedure is depicted in Fig. 2, alongside the input and reconstructed log-magnitude.

2.5. Synthesis Predictor

The synthesis predictor P is a component designed to convert the sequence of embeddings $y_{\text{enc}} \in \mathbb{R}^{N \times D}$ to a 2-dimensional vector y_{synth}, indicating the presence of synthetic speech by means of one-hot encoding, i.e.

$$P(y_{\text{enc}}, \Theta^{P}_{\text{dec}}) = \hat{y}_{\text{synth}}, \tag{17}$$

where Θ^{P}_{dec} denotes the set of model parameters, and

$$\arg\max\left(\hat{y}_{\text{synth}}\right) = \begin{cases} 1 & \text{F0,F1,F2 appear synthetic} \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

The predictor consists of a standard transformer with a class token used as input to a classification layer, which is a standard procedure for vision transformers [7].

Given a transformed sequence of patch embeddings $\hat{\mathbf{z}}^P$, defined by

$$\hat{\mathbf{z}}^P = T^P(\tilde{\mathbf{z}}^P + \mathbf{z}^P_{\text{pos}}), \tag{19}$$

where $\mathbf{z}^P_{\text{pos}}$ denotes a new set of 1D positional embeddings, and $\tilde{\mathbf{z}}^P \in \mathbb{R}^{(N+1) \times D^P}$ a projected input sequence with a class token, the first element $\hat{z}^P_0 \in \mathbb{R}^{D^P}$ of the transformed output sequence can be processed by a classification layer yielding the final output:

$$\hat{y}_{\text{synth}} = f\left(W_{\text{synth}} \cdot \hat{z}^P_0 + b_{\text{synth}}\right), \tag{20}$$

with W_{synth} and b_{synth} representing the weight matrix and bias vector of the linear classification layer, and $f(\cdot)$ its ac-

Table 1. Hyper-parameters of the SFAT-Net-3 transformers

Component	Global Params		MLP Blocks		Self-Attention Blocks	
	Depth	Embedding Size	Dimensions	Dropout	Number of Heads	Head Dimension
E_X – Magnitude Encoder	8	512	1024	0	8	64
E_Φ – Phase Encoder	8	512	1024	0	8	64
D^X_{dec} – Spectrogram Decoder	6	512	1024	0	8	64
D^F_{dec} – Multi-formant Decoder	4	512	1024	0	8	64
P – Synthesis Predictor	4	512	1024	0.1	6	64

tivation function. A more formal description of the predictor, including the use of projection matrices to accommodate changes of the inherent dimensions, can be found in the SFAT-Net-2 paper [6].

In our experiments, we decided to use a linear activation function at inference phase, and a sigmoid function with binary cross entropy (BCE) loss at training phase:

$$\begin{aligned} l_P(X, y_{\text{synth}} \mid \Theta_{\text{enc}}, \Theta^P_{\text{dec}}) = \\ = \text{BCE}\left(y_{\text{synth}}, P\left(E\left(X, \Theta_{\text{enc}}\right), \Theta^P_{\text{dec}}\right)\right) \end{aligned} \tag{21}$$

The BCE loss is defined as usual by

$$\text{BCE}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i), \tag{22}$$

with y being the ground truth, $\hat{y}$ the predicted value, and N the number of samples in the training dataset.

3. Experimental Setup

3.1. Dataset

Our experiments were carried out utilizing the ASVspoof 2019 Logical Access (LA) subset [29], which comprises *train*, *dev*, and *eval* partitions, each encompassing both genuine and artificially generated utterances. The *train* and *dev* ASVspoof partitions were created utilizing the same six synthesis algorithms (A01, ..., A06), whereas the *eval* partition showcases samples generated through thirteen distinct techniques (A07, ..., A19), mirroring an open-world scenario where unfamiliar synthesis algorithms produce unknown voices.

To refine the dataset, we eliminated any leading or trailing silence from the original content via the MarbleNet model for voice activity detection [11], and standardized the volume following the EBU recommendation on audio signal loudness [8], employing a sampling frequency of 16 kHz. Additionally, we merged the *train* and *dev* sections into a unified training set, reserving the *eval* segment for testing purposes.

The considerable disparity between genuine and artificially generated utterances, with ratios of 1:6 and 1:13 respectively, was addressed by oversampling the genuine recordings along with the application of a randomized starting offset to each training trial.

3.2. Training Parameters

In Table 1 we reported the training parameters for the transformers included by our model. To assess the distinct contribution of the multi-formant decoder, we maintained all hyper-parameters as suggested for SFAT-Net-2 unchanged. This includes using the same MLP dimension, number and dimension of heads, as well as the same depth – i.e., the number of alternating pairs of MSA and MLP blocks – that the original SFAT-Net architecture employed exclusively for F_0 decoding.

The STFT was computed using a window length of 32 msec and a hop size of 16 msec, and included a standard pre-emphasis filter with coefficient 0.97. Thus, X and Φ had size of $L = 128$ frames by $M = 256$ frequency bins, corresponding to 2.064 seconds of content. We applied a framing grid of 16×16 patches, resulting in a sequence length of $N = 128$.

The training started with a learning rate of 1e-4 and a batch size of 64, using Adam optimization [14] coupled with cosine annealing schedule [16] set to complete a full cycle ever 2 epochs. We used early stopping when the sum of the autoencoding and formant estimation losses obtained on 10% of the data kept for validation reached a plateau, which occurred after about 100 iterations.

3.3. Evaluation Metrics

The performance of SFAT-Net-3 will be evaluated by means of the Receiver Operating Characteristic (ROC) curve obtained by the model, and of the corresponding Area Under the Curve (AUC) and Equal Error Rate (EER).

Even though these metrics do not relate to a specific operating threshold, we believe that they still provide an excellent description of the overall performances: The shape of the ROC curve provides interesting insights on the behavior of the network for low alarm rates, i.e., for the conditions in which the network is likely to be applied for forensic examinations; the EER and AUC determine the inherent model

Table 2. Performance of SFAT-Net-3 vs baseline models

	SFAT-Net-3	SFAT-Net-2	SFAT-Net	Pure Tr.
EER (%)	**15.05**	16.59	17.51	20.26
AUC (#)	**0.932**	0.910	0.900	0.885

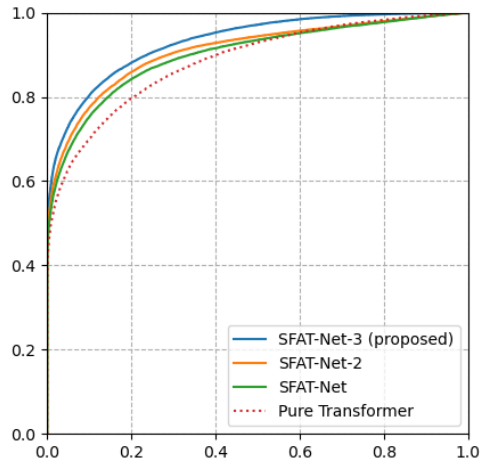


Figure 5. ROC curves of SFAT-Net-* on ASVspoof 2019 LA

Table 3. Performance of SFAT-Net-3 vs SOTA models

	SFAT-Net-3	AASIST	RawNet2	RawGAT-ST
EER (%)	**15.05**	17.10	20.67	23.00
AUC (#)	**0.932**	0.896	0.877	0.841

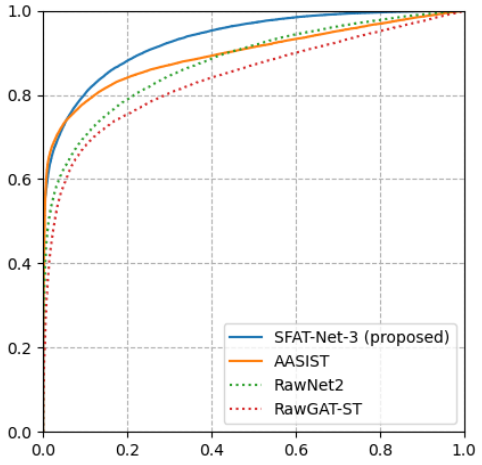


Figure 6. ROC curves vs SOTA models on ASVspoof 2019 LA

ability to discriminate the two classes in cases for which false alarms are more tolerable, e.g., when the results are applied for an initial screening of large content sets.

To ease the comparison with previous version of the model, we kept the conventions used in the SFAT-Net paper, and express EERs with percentages, and AUCs as pure numbers.

4. Evaluation

4.1. Contribution of the Multi-formant Decoder

In order to assess the distinct contribution of the multi-formant decoder, we compared the performance of SFAT-Net-3 against its predecessors.

Furthermore, we included in the evaluation a baseline model employing a pure transformer architecture, composed solely of the magnitude encoder E_X, the phase encoder E_Φ, and the predictor P. The training configuration and hyper-parameters of the transformers present in the resulting model were kept consistent with those detailed in Tab. 1 and Sec. 3. The results of this evaluation are reported in Tab. 2, and the corresponding ROC curves are depicted by Fig. 5.

SFAT-Net-3 is superior to all previous versions, achieving both a lower EER (15%) and a higher AUC (0.93) scores. From the ROC curve, we can observe how it consistently achieves a lower false alarm rate and higher recall across all possible operation points.

Crucially, all versions of the network surpass the pure transformer baseline, suggesting that focusing the attention on specific parts of the input through multi-task learning improves the overall generalization capabilities – which are essential to achieve a high score on the test partition of the ASVspoof 2019 LA dataset.

4.2. Comparison with the ASVspoof Baselines

In order to compare SFAT-Net-3 with the existing state of the art, we utilized a few baseline networks provided as open source by the ASVspoof committee, namely RawNet2 [25], RawGAT-ST [26] and AASIST [13].

RawNet2, at its core, is a SincNet architecture designed for speaker identification, which was fine-tuned for synthetic speech detection. The network learns a bank of pass-band filters, with widths and center frequencies determined by the training procedure – with the advantage of learning to focus on the parts of the input spectrogram which are relevant for the desired task [19, 25].

RawGAT-ST instead employs a graph attention network (GAT) on the spectro-temporal (ST) representation of the input speech. The ST representation is obtained using a SincNet encoder, i.e., by again learning a filterbank tailored to perform the desired task. Afterwards, the ST representation is converted to a fully-connected graph, and the edges of the graph are related to the attention computed between its nodes [26].

AASIST enhanced the graph-attention mechanism of its

Table 4. Performance breakdown on the spoofing algorithms of the ASVspoof 2019 LA test partition

	all	A07	A08	A09	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
EER (%)	15.05	0.00	3.20	1.63	23.49	14.14	26.49	26.44	1.14	21.46	0.10	23.45	9.92	0.07
AUC (#)	0.932	1.000	0.996	0.999	0.852	0.927	0.815	0.818	0.999	0.872	1.000	0.850	0.965	1.000

(a) SFAT-Net-3 (proposed model)

	all	A07	A08	A09	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
EER (%)	17.51	0.03	2.89	2.92	24.49	14.21	31.23	31.74	1.17	21.59	0.11	31.46	15.21	0.06
AUC (#)	0.900	0.999	0.995	0.995	0.811	0.924	0.736	0.729	0.999	0.847	0.999	0.734	0.913	0.999

(b) SFAT-Net (baseline in [5])

predecessor RawGAT-ST, by fusing the attention computed upon several graphs. While RawGAT-ST attended the entire graph in one shot, AASIST employs three graphs, one considering only connections within a time frame, a second considering only connections within a frequency band, and a last across both dimensions, as its predecessor [13].

All three models are provided by the respective authors with a pre-trained set of weights tailored to work with the original ASVspoof 2019 LA partitions. However, the pre-trained weights led to very poor performance akin to random guessing on our refined version of the dataset, in which we removed trailing and ending silence, and pre-normalized the loudness. Therefore, we retrained the three models using the same dataset as for SFAT-Net-3, with the same hyper-parameters described in the respective publications, until the classification loss obtained on 10% of data reserved for validation did not reach a plateau. The results obtained after retraining are reported in Tab. 3, and the corresponding ROC curves are depicted by Fig. 5.

In terms of EER (15%) and AUC (0.93) scores SFAT-Net-3 is superior to all baseline models provided by the ASVspoof organizers. This holds particularly true for both RawNet2 and RawGAT-ST, which are clearly outperformed. The AASIST architecture, however, is on par with RawNet2 for low alarm rates, but exhibits a lower recall for more lenient scenarios in which the false alarm rate can be 10% or higher.

4.3. Detailed outcome on ASVspoof 2019 LA

The performance of synthesis detection on the ASVspoof 2019 LA dataset was found to be noticeably uneven in the summary paper of the challenge [27].

Specifically, attacks A10, A13, and A18 were identified as significantly impairing Automatic Speaker Verification (ASV) performance while also proving challenging to detect. Additionally, the A17 attack emerged as the most elusive synthesis method to detect, although it presents a relatively low threat to ASV systems. Attacks A16 and A19 were classified as "known," meaning that synthetic content examples were provided in the training set, albeit with different voices and utterances.

In Tab. 4, we present the EER and AUC values of the ROC curves we obtained with SFAT-Net-3 for each individual synthesis algorithm in the test set, alongside the values obtained by the baseline SFAT-Net model. We can observe that the performance improved consistently across synthesis algorithm, especially for A12, A13, and A17, which were the most challenging algorithms for SFAT-Net – and for which the multi-formant estimation resulted in a reduction of the EER of about 10%.

4.4. Impact of lossy encoding

All the performance reported so far referred to uncompressed PCM content, having a sampling rate of 16 kHz.

These ideal conditions, however, do not match the analysis conditions expected for content retrieved from mobile devices or from the social media. Therefore, we investigated the performance of SFAT-Net-3 when faced with audio content which was lossy-encoded using AAC with several bitrates, as one would expect, e.g., for MP4 videos shared across the Internet.

The results obtained on lossy-encoded content are reported in Tab. 3, and the corresponding ROC curves are presented in Fig. 5.

We can observe a moderate degradation of the performance for bitrates equal to 32 kbps or above, which is promising for future deployment in real-world-scenarios.

Table 5. SFAT-Net-3 performance with lossy encoding

Encoding	PCM	AAC (kbps)			
		128	64	32	16
EER (%)	15.05	16.02	15.78	17.64	25.01
AUC (#)	0.932	0.922	0.923	0.909	0.831

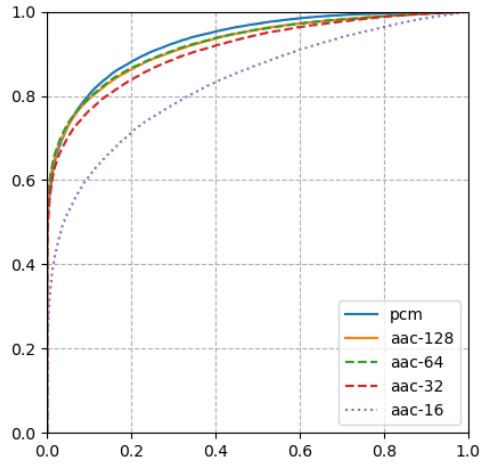


Figure 7. ROC curves of SFAT-Net-3 with lossy encoding

The network reliability drops significantly for content encoded with AAC at 16 kbps. The reason behind this behavior is the aggressive low-pass filtering applied by the encoding around 3.4 kHz, that removes more than half of the frequency range that the network would typically inspect for tracing formants and detecting synthetic speech.

Yet, considering that for bitrates above 32 kbps the performances are superior to those obtained by all ASVspoof baselines on PCM content, we deem SFAT-Net-3 to be generally reliable.

5. Conclusions and Outlook

In this paper we presented SFAT-Net-3, a novel multi-task transformer performing synthesis detection as a byproduct of estimating the phonetic formants F0, F1, and F2 of the input speech, and of reconstructing the entire input log-magnitude.

SFAT-Net-3 outperforms the previous version of the same architecture, and demonstrates superior or comparable performance to existing state-of-the-art baselines provided by the ASVspoof community. The model exhibits sufficient reliability even when processing input content encoded with AAC, and for input content encoded with at least 64 kbps, it was able to match the EER and AUC performance that state-of-the-art models achieve for PCM encoded content.

In future work, we aim to explore more features derived from the phonetic domain, and to address the drop in performance due to the presence of aggressive low-pass filters, such as the one we encountered for AAC at 16 kbps. The issue might be mitigated, for instance, by adjusting the frequency range of the input signal to only consider frequencies up to 4 kHz, or by incorporating data augmentation techniques in our training procedure that include speech encoders commonly found in the consumer devices.

Furthermore, since the selection criteria for synthesis algorithms within ASVspoof preclude drawing definite conclusions about the results beyond mere numerical performance, we plan to evaluate SFAT-Net-3 and existing state-of-the-art models using additional datasets, such as ODSS [30] and TIMIT-TTS [22], which are designed to mitigate such interpretation challenges.

Lastly, while SFAT-Net-3 has introduced significant performance improvements over its predecessors, it has not fully eradicated detection challenges associated with the most critical algorithms in the ASVspoof evaluation partition. To enhance predictive capabilities, further investigation into features drawn from the phonetic domains is needed, aiming to address these lingering weaknesses.

Acknowledgments

This paper was supported by the BMBF SpeechTrust+ project (grant no 13N16267), and by the EU Horizon Europe vera.ai project (grant no. 101070093).

Additionally, we would like to thank Kristina Tomić (University of Niš), Katharina Klug (University of York), and Sebastian Musche (LKA Baden-Württemberg) for their precious insights on the phonetics domain, and all the anonymous reviewers for their valuable feedback.

References

[1] Paul Boersma and David Weenink. Praat: doing phonetics by computer [Computer program], 1992–2022. Version 6.1.09. 3, 4

[2] Kalina Bontcheva, Symeon Papadopoulous, Filareti Tsalakanidou, Riccardo Gallotti, Lidia Dutkiewicz, Noémie Krack, Denis Teyssou, Francesco Severio, Jochen Spangenberg, Ivan Srba, Patrick Aichroth, Luca Cuccovillo, and Luisa Verdoliva. *Generative AI and Disinformation: Recent Advances, Challenges, and Opportunities*. European Digital Media Observatory, 2024. 1

[3] European Commission. *Proposal for a regulation of the European parliament and of the council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain Union Legislative Acts*. Publications Office of the European Union, 2021. 1

[4] Luca Cuccovillo, Christoforos Papastergiopoulos, Anastasios Vafeiadis, Artem Yaroshchuk, Patrick Aichroth, Konstantinos Votis, and Dimitrios Tzovaras. Open challenges in synthetic speech detection. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, Shanghai, China, 2022. 1

[5] Luca Cuccovillo, Milica Gerhardt, and Patrick Aichroth. Audio spectrogram transformer for synthetic speech detection via speech formant analysis. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, Nürnberg, Germany, 2023. 1, 3, 7

[6] Luca Cuccovillo, Milica Gerhardt, and Patrick Aichroth. Audio transformer for synthetic speech detection via formant magnitude and phase analysis. In *IEEE International*

Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, South Korea, 2024. in press. 2, 3, 5

[7] 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. In *International Conference on Learning Representations (ICLR)*, Vienna, Austria, 2021. 2, 4

[8] EBU. R128-2020: Loudness normalization and permitted maximum level of audio signals, 2020. 5

[9] Europol. *Facing reality? Law enforcement and the challenge of deepfakes, an observatory report from the Europol Innovation Lab*. Publications Office of the European Union, 2022. 1

[10] Augustine H. Gray and David Y. Wong. The Burg algorithm for LPC speech analysis/synthesis. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(6):609–615, 1980. 4

[11] Fei Jia, Somshubra Majumdar, and Boris Ginsburg. MarbleNet: Deep 1D time-channel separable convolutional neural network for voice activity detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6818–6822, Toronto, ON, Canada, 2021. 5

[12] Keith Johnson. *Acoustic and Auditory Phonetics*. Cambridge University Press, 2011 [1997]. 3d edition. 2

[13] Jee-weon Jung, Hee-Soo Heo, Hemlata Tak, Hye-jin Shim, Joon Son Chung, Bong-Jin Lee, Ha-Jin Yu, and Nicholas Evans. AASIST: Audio anti-spoofing using integrated spectro-temporal graph attention networks. In *arXiv preprint arXiv:2110.01200*, 2021. 1, 6, 7

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 2015. 5

[15] John Laver. *The Phonetic Description of Voice Quality*. Cambridge University Press, 1980. 2

[16] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, Toulon, France, 2017. 5

[17] Nicolas M. Müller, Karla Pizzi, and Jennifer Williams. Human perception of audio deepfakes. In *ACM International Workshop on Deepfake Detection for Audio Multimedia (DDAM)*, pages 85–91, Lisboa, Portugal, 2022. 1

[18] Francis Nolan. *The phonetic bases of speaker recognition*. Cambridge University Press, 1983. 2

[19] Mirco Ravanelli and Yoshua Bengio. Speaker recognition from raw waveform with SincNet. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 1021–1028, Athens, Greece, 2018. 6

[20] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should I trust you?: Explaining the predictions of any classifier. In *ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1135–1144, San Francisco, CA, USA, 2016. 1

[21] Davide Salvi, Paolo Bestagini, and Stefano Tubaro. Towards frequency band explainability in synthetic speech detection. In *European Signal Processing Conference (EUSIPCO)*, pages 620–624, Helsinki, Finland, 2023. 1

[22] Davide Salvi, Brian Hosler, Paolo Bestagini, Matthew C. Stamm, and Stefano Tubaro. TIMIT-TTS: A text-to-speech dataset for multimodal synthetic media detection. *IEEE Access*, 11:50851–50866, 2023. 8

[23] Davide Salvi, Temesgen Semu Balcha, Paolo Bestagini, and Stefano Tubaro. Listening between the lines: Synthetic speech detection disregarding verbal content. In *IEEE International Conference on Acoustics, Speech and Signal Processing Workshops (ICASSPW)*, Seoul, South Korea, 2024. in press. 1

[24] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, Venice, Italy, 2017. 1

[25] Hemlata Tak, Jose Patino, Massimiliano Todisco, Andreas Nautsch, Nicholas Evans, and Anthony Larcher. End-to-end anti-spoofing with RawNet2. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6369–6373, Toronto, ON, Canada, 2021. 1, 6

[26] Hemlata Tak, Jee weon Jung, Jose Patino, Madhu Kamble, Massimiliano Todisco, and Nicholas Evans. End-to-end spectro-temporal graph attention networks for speaker verification anti-spoofing and speech deepfake detection. In *Automatic Speaker Verification and Spoofing Countermeasures Challenge (ASVspoof)*, pages 1–8, 2021. 1, 6

[27] Massimiliano Todisco, Xin Wang, Ville Vestman, Md Sahidullah, Héctor Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi Kinnunen, and Kong Aik Lee. Asvspoof 2019: Future horizons in spoofed and fake audio detection. In *ISCA Annual Conference of the International Speech Communication Association (INTERPEECH)*, pages 1008–1012, Graz, Austria, 2019. 7

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *International Conference on Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, Long Beach, CA, USA, 2017. 1, 2

[29] Xin Wang, Junichi Yamagishi, Massimiliano Todisco, Héctor Delgado, Andreas Nautsch, Nicholas Evans, Md Sahidullah, Ville Vestman, Tomi Kinnunen, Kong Aik Lee, *et al*. ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech. *Computer Speech & Language*, 64, 2020. 5

[30] Artem Yaroshchuk, Christoforos Papastergiopoulos, Luca Cuccovillo, Patrick Aichroth, Konstantinos Votis, and Dimitrios Tzovaras. An open dataset of synthetic speech. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, Nürnberg, Germany, 2023. 8