DTF-AT: Decoupled Time-Frequency Audio Transformer for Event Classification

Tony Alex[1], **Sara Ahmed**[1,2], **Armin Mustafa**[1,2], **Muhammad Awais**[1,2], **Philip JB Jackson**[1,2]

[1]Surrey Institute for People-Centred AI, University of Surrey, Guildford, GU2 7XH, UK
[2]Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey
t.alex@surrey.ac.uk, sara.atito@surrey.ac.uk, armin.mustafa@surrey.ac.uk, muhammad.awais@surrey.ac.uk, p.jackson@surrey.ac.uk

Abstract

Convolutional neural networks (CNNs) and Transformer-based networks have recently enjoyed significant attention for various audio classification and tagging tasks following their wide adoption in the computer vision domain. Despite the difference in information distribution between audio spectrograms and natural images, there has been limited exploration of effective information retrieval from spectrograms using domain-specific layers tailored for the audio domain. In this paper, we leverage the power of the Multi-Axis Vision Transformer (MaxViT) to create DTF-AT (**D**ecoupled **T**ime-**F**requency **A**udio **T**ransformer) that facilitates interactions across time, frequency, spatial, and channel dimensions. The proposed DTF-AT architecture is rigorously evaluated across diverse audio and speech classification tasks, consistently establishing new benchmarks for state-of-the-art (SOTA) performance. Notably, on the challenging AudioSet 2M classification task, our approach demonstrates a substantial improvement of 4.4% when the model is trained from scratch and 3.2% when the model is initialised from ImageNet-1K pretrained weights. In addition, we present comprehensive ablation studies to investigate the impact and efficacy of our proposed approach. The codebase and pretrained weights are available on https://github.com/ta012/DTFAT.git

Introduction

The field of Audio pattern recognition has seen massive progress due to the advent of deep learning. From sequential models to convolutional neural networks (CNNs) (Hershey et al. 2017; Kong et al. 2020) and now to transformers (Gong, Chung, and Glass 2021a; Koutini et al. 2022; Chen et al. 2022a), the incremental progress in neural network architectures has been reflected in the performance of audio classification tasks. Convolutional neural networks have been the go to network for audio classification up until the introduction of Transformers. Recently Transformers were shown to outperform CNN-based networks.

Both convolutional and transformer (self-attention) layer exhibit intrinsic learning capabilities that differentiate them significantly, making the replacement of one with the other difficult. The use of data-independent fixed kernels makes convolution less prone to overfitting and better in generalisation even with small datasets due to their strong prior of inductive bias (Dai et al. 2021). Conversely, the data-dependent attention weights, help transformer layer in learning more complex interactions compared to convolutional layers with the added risk of overfitting. Furthermore, large receptive fields in transformers give them higher modelling capacity than convolutional layers (Dai et al. 2021). Thus convolutional neural networks (CNNs) tend to work well even with small datasets, where transformers tend to fail, where as transformers can outperform convolutional networks on large datasets. Additionally, the success of various hybrid vision architectures (Dai et al. 2021; Tu et al. 2022; Fan et al. 2021) serves as a testament to synergistic learning capabilities brought by convolutions and self-attention. The datasets in audio vary significantly in sizes, e.g., ESC50 dataset has 2000 audio files while Audioset2M has around 2 million audio files. Therefore, it is important to have a network that works well across varying dataset sizes and audio file durations.

A common trend in audio deep learning research is to adopt successful vision networks, because of the 2D nature of the most commonly used audio input format (mel-spectrogram). Although its 2-D nature similar to that of an image facilitates it to be processed as one by the vision networks, it is important to note that the information distribution across an audio spectrogram is distinct from that in

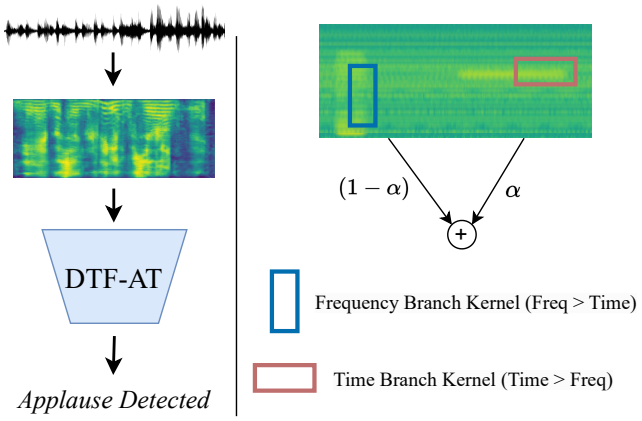

Figure 1: Overview of the audio classification task (left) and Time-Frequency Decoupling (right), α: branch multiplier.

an image. For example, in an image, an object can appear anywhere within the frame without constraints, whereas in an audio spectrogram, the representation of an object is bounded by its frequency characteristics. Hence information extraction techniques for images may not be optimal for audio spectrograms. In this paper, we investigate effective feature learning given the asymmetry of information which is inherent in audio spectrogram across the time and frequency axes. Specifically, we propose an audio architecture that extracts information along the time and frequency axes independently.

Our main contributions in this work are

- Time-Frequency decoupling technique for effective audio feature extraction.
- A transformer block for audio, that combines spatially decoupled information along time and frequency axis along with local and global context learning by self-attention layers.
- SOTA performance across various audio datasets with an improvement of 4.4% when the model is trained from scratch and 3.2% when the model is initialised from ImageNet-1K pretrained weights for AudioSet full set.

Previous Work

Notable CNN-based approaches are the investigations carried out by Hershey et al. (2017) and Kong et al. (2020). Hershey et al. (2017) delve into the utilization of well-established CNN architectures, while Kong et al. (2020) explore a range of factors influencing CNN-based audio pattern recognition, such as optimal input format, transferability etc. Gong, Chung, and Glass (2021b), proposed a hybrid audio architecture that combines convolutional layers at the initial stage with subsequent attention layers and various training techniques.

AST(Gong, Chung, and Glass 2021a) explores attention-based network inspired by the vision transformer ViT(Dosovitskiy et al. 2020). PaSST(Koutini et al. 2022) introduced the idea of dropping certain patches(Patchout) for the AST during training and disentangling the positional encoding into time and frequency components. Patchout also acts as a regularizer during the training improving the performance further. Inspired by the Swin(Liu et al. 2021) Chen et al. (2022a) proposed HTS-AT. Input patches are ordered in a time-frequencey-window order to create a tensor of resolution (256,256), which is suitable for operations such as window partitioning with window size (8,8). A convolutional module(Token Semantic Module) is used to generate the classification outputs, which further improved the performance. Li et al. (2022) adapted vision transformer Muliscale V2 to the audio domain in their work titled MAST(Zhu and Omar 2023). Zhang et al. (2022) introduced audio architecture tailored for audio by generating time and frequency embedding from spectrograms, followed by temporal and frequency attention blocks, subsequently connecting to a classification head. Their approach, when trained from scratch, demonstrates the ability to outperform AST on the ESC50 dataset.

Transformer architectures are known to improve performance when trained with self-supervision objectives such as contrastive learning (Chen et al. 2020), masked image modelling (Atito, Awais, and Kittler 2021), image reconstruction etc. As per our current knowledge, BEATs(Chen et al. 2022b), which utilize acoustic tokenizers for SSL, currently holds the state-of-the-art performance on the Audioset fullset (non-ensembling). Other approaches utilizing masking, reconstruction include SSAST (Gong et al. 2022), MAE-AST (Baade, Peng, and Harwath 2022), CAT (Liu et al. 2023), ASiT (Atito et al. 2022) and Masked Autoencoders that Listen (Huang et al. 2022). Moreover, contrastive learning methods consist of CLAR (Al-Tahan and Mohsenzadeh 2021) and COLA (Saeed, Grangier, and Zeghidour 2021), while distillation-based approaches involve BOYL-Audio (Niizumi et al. 2021) and ATST (Li and Li 2022). In this work, we will not be considering approaches using self-supervision objectives. Instead we plan to do that in our future work exploring the potential enhancements that self-supervision can introduce to our network.

Methodology

In this paper, we propose a novel audio transformer block that compartmentalizes the audio feature learning into time and frequency, and seamlessly incorporates both local and global contextual elements. Specifically, we adapted and modified the MaxViT framework to suit audio processing, demonstrating its robustness as an audio transformer, all of which are briefly detailed in the following sections.

Building upon this foundation, we take it a step further and propose the concept of time-frequency decoupling, which involves learning time and frequency based concepts via independent branches for effective feature learning.

Over the next sections, we explore various constituents of our architecture and consolidate them in the fourth section.

Preliminaries

We based our architecture on the vision transformer MaxViT (Tu et al. 2022). The fundamental building block of MaxViT comprises of a MBConv block (Howard et al. 2017), a window/block attention block, and a grid attention block. Having window and grid attention facilitates both local and global interaction in every transformer block. By combining MBConv with the attention mechanisms, MaxViT enhances its generalisation ability and trainability.

Time-Frequency Decoupling

Time-Frequency decoupling aims to bifurcate the aspects of learning that can be brought by frequency and time-based information. For instance, if we consider a 10-second audio clip, details about a musical instrument are more local in the frequency axis than in the time axis. Similarly, phenomenon such as silence between sounds is local in time.

To that end, we propose to replace the convolutional layer in MaxViT with a two separate convolutional branches to enable the decoupled learning. The first branch focuses on time with a comparatively larger receptive field spanning the time axis and the second branch focuses on frequency with a

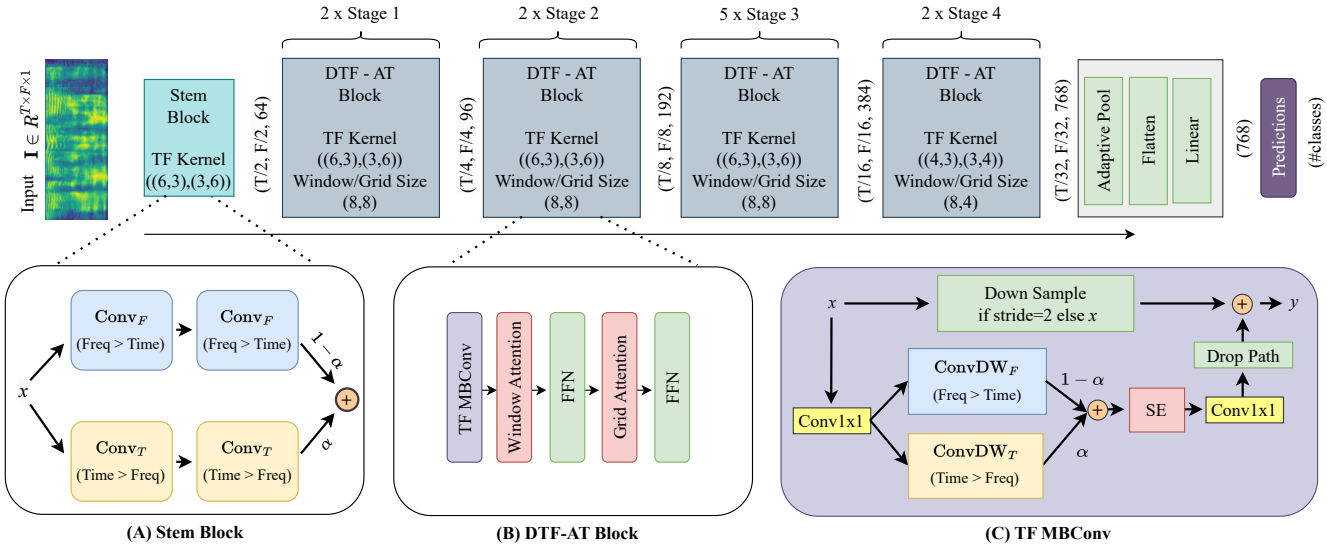

Figure 2: DTF-AT Architecture. T: Time axis, F: Frequency axis. (A) Stem Block: Input spectrogram is processed independently using branches one with kernels that span more on the time axis and vice versa as explained in Equation 2. (B) DTF-AT Block with Time-Frequency Decoupled MBConv Block (Equation 1 and 3).

comparatively larger receptive field spanning the frequency axis. The the basic concept of the proposed approach is depicted in Figure 1 (right).

Precisely, for an audio input sample $\mathbf{I}$, the processing in the TF decoupled convolutional block is as follows:

$$
\begin{aligned}
\bar{\mathbf{x}} &= \begin{cases} \text{DownSample}(\mathbf{I}), & \text{if stride} = 2 \\ \mathbf{I}, & \text{otherwise} \end{cases} \\
\mathbf{x} &= \text{BN–Act}(\ \text{Conv1x1}(\ \text{BN}(\mathbf{I})\ )\ ) \\
\mathbf{x} &= \alpha \times \text{ConvDW}_T(\mathbf{x}) + (1-\alpha) \times \text{ConvDW}_F(\mathbf{x}) \quad (1) \\
\mathbf{x} &= \text{SE}(\ \text{BN–Act}(\mathbf{x})\ ) \\
\mathbf{x} &= \text{Conv1x1}(\mathbf{x}) \\
\mathbf{x} &= \text{DropPath}(\mathbf{x}) + \bar{\mathbf{x}}
\end{aligned}
$$

where DownSample denotes the down-sampling performed across time and frequency axes in the first transformer block of all stages. ConvDW_T and ConvDW_F stand for depth-wise convolution with kernel spanning time axis more than frequency and vice versa. $\text{Conv1x1}, \text{BN}, \text{Act}, \text{SE}$, and DropPath denote 1×1 convolution, batch norm, activation layer, squeeze and excitation block (Hu, Shen, and Sun 2018), and drop path (Huang et al. 2016) operation, respectively. For specifics regarding layers, kernel sizes, and feature map sizes, refer to Figure 2.

In this arrangement, the *time branch is more local in frequency*, allowing it to concentrate on time-dependent concepts with relatively low variations in frequency over time. On the other hand, the *frequency branch is more local in time*, enabling it to handle variations in frequency over a very short period in time (though this is minimal in the initial layers due to the smaller kernel size). It's crucial to note that this intuition only holds when a minimum receptive field is present for frequency in the time branch and vice versa. That is given a fixed number of parameters $3x$ for the convolutional kernel, kernel of size $(x, 2x)$ and $(2x, x)$ would be a favourable choice when contrasted with $(1,3x)$ and $(3x,1)$. Further discussion with empirical evidence is present in the ablation.

The fundamental concept behind TF decoupling involves considering more data along either time or frequency axes while localising the other. Consequently, the TF decoupling approach can also be implemented in the local self-attention operations, such as window attention. While we briefly address this aspect in our ablations, a comprehensive exploration of parameters such as window size, attention heads for branches, etc is essential to effectively implement this in window attention. In this paper, however, our emphasis remains on TF decoupling within convolutions.

Time-Frequency Decoupled Stem Block

In addition to introducing TF decoupling in the convolutional layers of MaxViT, we extend this concept to the stem of the architecture as well. *Convolutional* stems have demonstrated enhanced optimisation stability and performance compared to *patchify* stems in the vision domain (Xiao et al. 2021). In contrast to prior audio transformers (Gong, Chung, and Glass 2021a; Chen et al. 2022a) employing *patchify* stem, we use a TF decoupled *convolutional* stem in our network as shown in Figure 2(A). We postulate that the time and frequency stem branches support the effective processing of time-frequency-related concepts extracted from the raw spectrogram. Specifically, the input $\mathbf{I}$ is processed in the stem as follows:

$$
\begin{aligned}
\mathbf{x_t} &= \text{Conv}_T(\text{BN–Act}(\text{Conv}_T(\mathbf{I})) \\
\mathbf{x_f} &= \text{Conv}_F(\text{BN–Act}(\text{Conv}_F(\mathbf{I})) \quad (2) \\
\mathbf{x} &= \alpha \times (\mathbf{x_t}) + (1-\alpha) \times (\mathbf{x_f})
\end{aligned}
$$

where Conv_T, Conv_F, stands for convolution with kernel spanning time axis more than frequency and vice versa. For specifics regarding layers, kernel sizes, and feature map sizes, refer to Figure 2.

Local-Global Interaction Block

Inspired from the vision transformer MaxViT (Tu et al. 2022), our audio transformer block consists of a time-frequency decoupled convolutional block (detailed in the previous sections), window self-attention block, and a grid self-attention block (Equation 3). For input $\mathbf{I}$, window/grid attention computation is as follows,

$$\begin{aligned}
\mathbf{x} &= \text{Window/Grid_Partition}(\text{LN}(\mathbf{I})) \\
\mathbf{x} &= \text{RelativeSelfAtten}(\mathbf{x}) \\
\mathbf{x} &= \text{Reverse_Window/Grid_Partition}(\mathbf{x}) \\
\mathbf{x} &= \text{DropPath}(\mathbf{x}) + \mathbf{I} \\
\mathbf{x} &= \text{DropPath}(\text{MLP}(\text{LN}(\mathbf{x})) + \mathbf{x}
\end{aligned} \tag{3}$$

where LN and MLP stand for layer norm and multi-layer perceptron, respectively, and the relative self attention block is defined as follows:

$$\text{RelativeSelfAtten}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}} + B\right)V \tag{4}$$

The convolutional block helps in learning time and frequency concepts for the following blocks to process, improving the generalisation ability of the network and learning local interactions. whereas window self-attention, a type of self-attention within local neighbourhood helps in learning complex local interactions. Finally, grid attention facilitates global interaction efficiently via dilated self-attention.

The combination of all the 3 blocks enables the network to learn both local and global information in every block of the network. Although both convolutions and window attention foster local interaction, convolutions achieve this by the use of fixed kernel that focuses on the relative position of tokens/pixels than its values which helps it in having better generalisation ability and properties such as translation equivalence, whereas in window attention the attention weight is computed from the data itself. This helps window self-attention to capture more complex local interactions with the possibility of overfitting. Also, as the time-frequency components are predominantly local in a spectrogram, the audio feature learning would gain significant advantages from local interactions rather than global ones. Further discussion is present in the ablation analysis. Therefore, our audio transformer block is designed to enable a high degree of local interactions via convolution and window attention.

The sequential placing of window attention and grid attention one after the other helps all tokens to interact with every other token in an efficient way. For example, consider a window size of (3×3) as in Figure 3(left) token 1 interacts with all other red tokens, then with other tokens in the spectrogram in a dilated fashion (Figure 3(right)). This effectively facilities global token interaction even though most of them are not directly interacting.

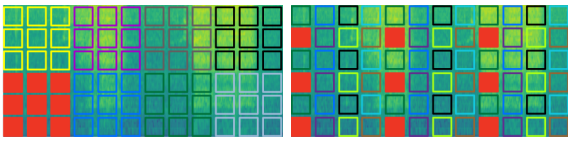

Figure 3: Token interaction in window and grid attention blocks (window/grid size:(3×3), Time:12, Frequency:6). In window attention(left) tokens inside the window of window_size (3×3) (same colour) interact, whereas in grid attention(right) tokens with a gap of (T/grid_size $\times$ F/grid_size) (4×2) interact (same colour). One set of interacting tokens each in window and grid blocks are coloured in red for illustration purposes.

Bringing All Together

Our transformer block (Figure 2 (B)) consists of a TF decoupled MBConv(Inverted Residual Block) Equation 1 (Figure 2 (C)), a window attention block, and a grid attention block. A TF decoupling-enabled Stem network Equation 2 (Figure 2 (A)) is used to process the raw log Mel filter bank(fbank) input for the proceeding transformer blocks. In the variant where TF decoupling is not implemented a single branch Stem and MBConv blocks with kernels of size (3×3) were used.

The network consists of 4 stages(11 DTF-AT blocks) stacked sequentially following the stem block. The resolution and number of channels are modified in a hierarchical fashion (Figure. 2). Input spectrogram resolution is downsampled by half using strided convolution and the number of channels is brought to 64 in the stem network. After the stem network, the feature map is processed in a hierarchical fashion as explained in Figure 2. Class probabilities are calculated using a linear layer, after performing global pooling on the output of stage 4. A window size (we kept window size equal to grid size) of (8×8) is used until stage 3 and (8×4) in stage 4. This makes tokens inside a local window of (8×8) interact in the window attention block and tokens with a gap of (feature map size/window size) between them interact in the grid attention block (Figure 3), facilitating both local and global attention.

Experiments

In this section, we evaluate the proposed architecture in various benchmark audio datasets, followed by ablation experiments to assess the various choices made during network development. Our proposed architecture demonstrated superior performance compared to existing approaches across various datasets, including Audioset (Gemmeke et al. 2017), ESC50 (Piczak 2015), and Speech Commands V2 (Warden 2018) datasets. We assigned names to the architecture variants that readily convey the extent of TF decoupling in them.

AudioSet

Dataset. AudioSet (Gemmeke et al. 2017) consists of audio files downloaded from YouTube. There are 3 subsets with 527 labels commonly used in experiments namely full

	balanced set mAP	full set mAP	#Params (Million)	MACs (GMac)
Training from ImageNet-1K Pretrained Weights				
AST (Gong, Chung, and Glass 2021a)	0.347±0.001	0.459±0.0	88.1	103.4
AST (Our Eval data)	-	0.460	88.1	103.4
PaSSAT (Koutini et al. 2022)	-	0.471	-	-
HTSAT (Chen et al. 2022a)	-	0.471	31.0	-
HTSAT (Our Eval data)	-	0.470	31.0	-
MAST† (Zhu and Omar 2023)	0.314	0.390	51.3	25.6
No TF (Ours)	0.349±0.001	0.483±0.000	68.6	29.4
Stem TF (Ours)	0.350±0.001	0.484±0.001	68.7	33.1
Full TF (Ours)	**0.355±0.001**	**0.486±0.001**	69.0	33.3
Training from Scratch (Random Initialisation)				
PANNs (Kong et al. 2020)	0.278	0.439	-	-
AST	0.148	0.366	88.1	103.4
HTSAT	-	0.453	31.0	-
No TF (Ours)	0.177±0.002	0.468±0.001	68.6	29.4
Full TF (Ours)	0.187±0.001	**0.473±0.000**	69.0	33.3

† Comparatively small training set

Table 1: AudioSet performance comparison with the previous approaches. Full TF: TF decoupling across the entire architecture, Stem TF: TF decoupling in stem block only, No TF: No TF decoupling.

set (∼2M audio files), balanced set (∼20k), and evaluation set (∼20k). Since the AudioSet data is downloaded from YouTube directly, videos get deleted and the available dataset decreases in size over time. In this work, we employed the downloaded copy provided by PANNs (Kong et al. 2020) which contains 1.93 million samples from the original dataset. To the best of our knowledge, the number of audio files in our data set is slightly lower compared to previous works, such as AST (Gong, Chung, and Glass 2021a). For a fair comparison with the state-of-the-art approaches, in Table 1, we report the performance of the best models shared by AST (Gong, Chung, and Glass 2021a) and HTSAT (Chen et al. 2022a) on our evaluation set along with their reported performances, despite that these models are trained on a full set with more than 2 million audio files.

Training Details. We converted audio files of 10 seconds duration and 32kHz sample rate to 128-dimensional Mel filterbank (fbank) features resulting in an input shape of 1024×128. For training, we used the same pipeline as AST (Gong, Chung, and Glass 2021a), with minor changes such as performing learning rate update after every 500k audio files for full set training. Mixup (Tokozume, Ushiku, and Harada 2018) with 0.5 ratio and Spectrogram masking (Park et al. 2019) (max time mask length of 192 and max frequency mask length of 48 bins) were used for data augmentations. We used an initial learning rate of $5e^{-4}$ for both full set and balanced set. The learning rate is updated by multiplying with a factor of 0.5 for the full set and 0.1 for the balanced set at certain validation steps using Multi-step learning rate scheduler. The models are trained with AdamW optimiser (Loshchilov and Hutter 2019) and binary cross entropy loss function. We employed Mean average precision (mAP) as the evaluation metric and ran our experiments for 3 times with different random seeds and the mean and standard deviation of the best epochs are reported.

We conducted experiments using two different settings. In the first setting, we trained the models entirely from scratch, i.e. weights are randomly initialised. In the second setting, we initialised our models from ImageNet-1K (Deng et al. 2009) pretrained weights. For the balanced set, the models are trained for 50 epochs using 32 batch size. As for the full AudioSet dataset, the models are trained with batch size of 64 for 10 and 8 epochs in the aforementioned settings, respectively.

Results. As shown in Table 1, all architecture variants based on the extent of TF decoupling perform better than the state-of-the-art architectures with the best one (Full TF) giving a performance of 0.473 mAP which is an improvement of 4.4% over the state-of-the-art (SOTA) approach, i.e. HTSAT, when the models are trained from scratch. Furthermore, we show an improvement of 3.2% when the model is trained from ImageNet-1K pretrained weights over the SOTA methods. Also, it is interesting to note that our Full TF network, trained from scratch, surpasses the SOTA approach initialised with ImageNet-1K weights. In addition, we performed a straightforward ensemble technique by aggregating the best models trained from scratch across three random seeds. Remarkably, this simple ensembling strategy yielded a better mAP of 0.498, highlighting the substantial room for further improvement.

While the primary focus of this paper remains on supervised learning, it is worth noting that our Full TF model performance with mAP of 48.6 achieved similar performance as the current SOTA self-supervised approach, namely BEATs (Chen et al. 2022b). As transformer architectures are known to improve performance when trained with self-supervision objectives, we are planning to investigate the potential performance enhancements that self-supervision objectives could bring to our existing architecture in our future works.

ESC50 and Speech Commands V2

Datasets. ESC50 is a collection of 2000, 5-second audio files with 35 classes. The Speech Commands V2 consists of $84,843$ audio files in the train set, $9,981$ files in the validation set, and $11,005$ files in the evaluation set, each containing a spoken word of duration 1 second and spanning 35 classes.

Training Details. We follow the same experiment protocol followed by AST. For ESC50, the data is divided into 5 folds. The model is then trained five times, each time using a different fold as the evaluation set and the remaining four folds as the training set. After completing the five training iterations, the performance metrics(accuracy) are collected from each evaluation, and the average performance across all 5 folds is calculated. We repeat this experiment 3 times with different random seeds and the mean and standard deviation of the performance metric accuracy is reported in Table 2. We converted audio files of 5 seconds duration and 32kHz sample rate to 128-dimensional Mel filterbank (fbank) resulting in an input shape of (512,128) and used the same training pipeline as AST (Gong, Chung, and Glass 2021a) with data augmentation such as Mixup (Tokozume, Ushiku, and Harada 2018)(mixup ratio=0.5) and Spectrogram masking (Park et al. 2019) (max time mask length:96 bins, max frequency mask length: 24 bins). During training, we experimented with two different initialisation methods: ImageNet-1K and AudioSet full set initialisation. The network is trained for 50 epochs with batch size 64 and learning rate $5e{-}4$ (ImageNet-1K initialisation) and $5e{-}5$ (AudioSet full set initialisation).

For Speech Commands V2, the 1-second audio files with sample rate 16kHz are converted to 128-dimensional Mel filterbank (fbank) resulting in an input shape of (128,128) and trained using the same training pipeline as AST (Gong, Chung, and Glass 2021a). Similar to ESC50 dataset, the network is trained for 50 epochs using a batch size of 64 and a learning rate of $5e{-}4$ with ImageNet-1K initialisation, and $5e{-}5$ with AudioSet full set initialisation, employing data augmentations, such as time masking, random noise and mixup.

Results. As shown in Table 2, our proposed method achieved SOTA performance in both ESC-50 and SC-V2 datasets. For ESC-50 datatset, the best performing model is the one pretrained from AudioSet full set. Unlike ESC-50, ImageNet-1K pretrained initialisation worked better than AudioSet full set initialisation for Speech Commands V2 dataset which is in line with the observations made in AST (Gong, Chung, and Glass 2021a).

It's worth highlighting that prior study (Zhang et al. 2022) leveraging the time-frequency components of the spectrogram conducted their ESC-50 experiments with random initialization, achieving an accuracy of 57.24. In comparison, our network under the same conditions attained a significantly improved performance of 76.40 (refer to Table 3).

Compute Resources For AudioSet experiments, we employed a single Nvidia A100-80GB GPU, while for ESC50 and Speech Commands V2, we utilised one NVIDIA GeForce RTX 3090-24GB GPU, running on the Ubuntu OS and employing the PyTorch deep learning framework. The choice of our batch size is not limited by the GPU memory.

The time and frequency branching shouldn't be consuming extra time as they are expected to run in parallel, given enough memory, the parallelisation is not effective as expected in the PyTorch implementation. Hence, model forward time increases with the extent of TF decoupling.

	ESC50	SCV2
AudioSet fullset Pretrained		
AST	95.6±0.4	97.88±0.03
HTSAT	97.0±0.2	98.00±0.03
Full TF(Ours)	**97.5±0.06**	97.68±0.07
ImageNet-1K Pretrained		
AST	88.7±0.7	98.1±0.05
Full TF (Ours)	**89.2±0.04**	**98.3±0.03**

Table 2: ESC50 and Speech Commands V2 performance comparison with the previous approaches.

Ablation Studies

We performed comprehensive ablation experiments to validate the design of our network with empirical evidence obtained from datasets such as Audioset (Gemmeke et al. 2017) (balanced and full set), ESC50 (Piczak 2015), and Speech Commands V2 (Warden 2018), within our resource limitations.

Effect of Time-Frequency Decoupling. We conducted comprehensive ablation experiments to assess the impact of TF decoupling. We explored three variations of TF decoupling, including stem TF decoupling where decoupling is applied to the stem only, TF decoupling until stage 1 covering stem and first stage, and Full TF decoupling across the entire architecture. In Table 3, we present results for AudioSet (balanced and full set), ESC50, and Speech Commands V2 datasets. Since we employ three significant digits, subtle variations are not discernible. Nonetheless, the table demonstrates the performance enhancement associated with the extent of decoupling. Notably, TF decoupling consistently enhances performance across all experiments.

Effect of ImageNet-1K Pretraining. To analyse the effect of initialising the model with ImageNet-1K pretrained weights, we experimented different datasets with all our variants with and without ImageNet-1K initialisation. Our experiments showed that ImageNet-1K pretraining helps in improving the performance as shown in Table 3.

Effect of Kernel Size. The number of parameters with and without decoupling differs, albeit by a marginal amount as shown in Table 1. To see if the performance improvement is due to the extra parameters and computation, we created a network without TF decoupling but with kernel size (6×6), which is the same number of parameters $(6 \times 6 = 36)$ as in decoupling $(6 \times 3 + 3 \times 6 = 36)$. Table 4 shows the performance improvement of decoupling not due to the slight improvement in parameters. Also, we experimented with different kernel sizes to find the optimal kernel size for decoupling (Table 4). Due to the variation in kernel size and feature map resolution of our hierarchical transformer, we

	balanced set	full set	ESC50	SCV2
ImageNet-1K Pretrained				
No TF	0.349	0.483	87.99	98.20
Stem TF	0.350	0.484	88.50	98.12
TF Till Stage 1	0.352	0.484	88.55	98.27
Full TF	0.355	0.486	89.19	98.30
Scratch (No Pretraining)				
No TF	0.177	0.468	75.30	97.78
TF Till Stage 1	0.184	0.470	75.95	97.85
Full TF	0.187	0.473	76.40	97.87

Table 3: Effect of Time-Frequency Decoupling and Effect of ImageNet-1K pre-training.

Time	Frequency	balanced set
ImageNet-1K Pretrained		
(6×3)	(3×6)	0.352
(9×2)	(2×9)	0.342
(18×1)	(1×18)	0.314
No TF decoupling - kernel size 6×6		0.344

Table 4: Effect of kernel size in Time and Frequency branches. All the models are trained from ImageNet-1K pretrained weights and TF decoupling till stage 1.

performed TF decoupling till block 4 (stage 1) for all these experiments. As we discussed earlier, among the kernel sizes $((18\times1),(1\times18))$, $((9\times2),(2\times9))$ and $((6\times3),(3\times6))$ (number of parameters:36) ((6,3),(3,6)) seems to perform better. This proves our hypothesis that a minimum receptive field should be present for frequency in the time branch and vice versa for effective local interactions.

TF Decoupling in Convolution vs Window attention block. As the idea of decoupling is to bifurcate time based and frequency based concept learning, it can be implemented in either the convolutional or the attention block such as window attention. To evaluate this idea, we ran two sets of experiments were we trained our framework on the full set of Audioset for 5 epochs with TF decoupling in the convolutional layers and in the window attention layers. The former showed superior performance, achieving an mAP of 0.450 compared to 0.443. With proper fine-tuning of window sizes, attention heads, and other parameters through extensive research, we anticipate that the window decoupling approach will also yield promising results. In this paper, we limit the decoupling to convolutional layers.

Impact of the Convolutional, Window Attention, and Grid Attention in Transformer Block. This ablation experiment is done without ImageNet-1K pretrained weights as we are changing the architecture significantly, and trained the model for only 5 epochs in each experiment. The ablation strategy is explained below,

- Remove TF decoupled MBConv blocks.
- Replace all window attention blocks with grid attention blocks.
- Replace all grid attention blocks with window attention blocks.

	full set
Scratch (No Pretraining)	
Full TF	0.450
Remove TF MBConv	0.392
Replace window with grid	0.436
Replace grid with window	0.449

Table 5: Transformer block components.

	balanced set
ImageNet-1K Pretrained	
Scalar (α, 1 - α)	0.355
Scalar (α, β)	0.349
Vector (α, 1 - α)	0.353
Vector (α, β)	0.353
Squeeze Excitation Weights	0.346

Table 6: Time-Frequency Decoupling Merging Strategies

As is evident from the performances in Table 5, convolution and window attention are two crucial components for audio feature learning. While grid attention is valuable, window attention complements it by enhancing its global attention capabilities. This validates the use of convolution, window attention, and grid attention in our transformer block.

Effect of Different merging techniques. We experimented with different merging techniques for time and frequency branches such as scalar (α, 1 - α), (α, β) and channel wise (α, 1 - α), (α, β). Among scalar weights, (α, 1 - α) seem to work better than (α, β). Regarding the use of channel-wise weight vs scalar weight, both seem to give similar performance. We also explored using the Squeeze Excitation layer in MBConv Block (Figure2 (C)) to obtain weights for merging by adding Squeeze Excitation to both branches for creating data dependent branch weights. We chose to go with scalar (α, 1 - α), as it is straightforward and easy to interpret the importance of time and frequency branches.

Regarding the values of branch weights in our network, we observed that the weights of both branches consistently range between 0.3 and 0.7. The frequency branch tends to receive slightly higher importance, while the importance of the time branches increases as the network depth grows.

Conclusion

As most audio transformers are inspired from vision transformers, they tend to deal with 2D audio spectrograms in a manner akin to natural images. In this paper, we proposed an audio transformer block that facilitates audio domain-specific time-frequency decoupling techniques with both local and global interaction for effective audio feature extraction. Extensive experiments demonstrate that the proposed TF decoupling exhibits promising performance, setting new benchmarks for state-of-the-art performance across various audio datasets.

Acknowledgments

Research was funded by EPSRC-BBC Prosperity Partnership 'AI4ME: Future personalised object-based media experiences delivered at scale anywhere' (EP/V038087/1). For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

References

Al-Tahan, H.; and Mohsenzadeh, Y. 2021. CLAR: Contrastive learning of auditory representations. In *International Conference on Artificial Intelligence and Statistics*, 2530–2538. PMLR.

Atito, S.; Awais, M.; and Kittler, J. 2021. Sit: Self-supervised vision transformer. *arXiv preprint arXiv:2104.03602*.

Atito, S.; Awais, M.; Wang, W.; Plumbley, M. D.; and Kittler, J. 2022. ASiT: Audio Spectrogram vIsion Transformer for General Audio Representation. arXiv:2211.13189.

Baade, A.; Peng, P.; and Harwath, D. 2022. Mae-ast: Masked autoencoding audio spectrogram transformer. *arXiv preprint arXiv:2203.16691*.

Chen, K.; Du, X.; Zhu, B.; Ma, Z.; Berg-Kirkpatrick, T.; and Dubnov, S. 2022a. HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 646–650. IEEE.

Chen, S.; Wu, Y.; Wang, C.; Liu, S.; Tompkins, D.; Chen, Z.; and Wei, F. 2022b. Beats: Audio pre-training with acoustic tokenizers. *arXiv preprint arXiv:2212.09058*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. Simclr: A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, 1597–1607.

Dai, Z.; Liu, H.; Le, Q. V.; and Tan, M. 2021. Coatnet: Marrying convolution and attention for all data sizes. *Advances in neural information processing systems*, 34: 3965–3977.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fan, H.; Xiong, B.; Mangalam, K.; Li, Y.; Yan, Z.; Malik, J.; and Feichtenhofer, C. 2021. Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6824–6835.

Gemmeke, J. F.; Ellis, D. P.; Freedman, D.; Jansen, A.; Lawrence, W.; Moore, R. C.; Plakal, M.; and Ritter, M. 2017. Audio Set: An ontology and human-labeled dataset for audio events. In *ICASSP*.

Gong, Y.; Chung, Y.-A.; and Glass, J. 2021a. Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*.

Gong, Y.; Chung, Y.-A.; and Glass, J. 2021b. PSLA: Improving Audio Tagging With Pretraining, Sampling, Labeling, and Aggregation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29: 3292–3306.

Gong, Y.; Lai, C.-I.; Chung, Y.-A.; and Glass, J. 2022. Ssast: Self-supervised audio spectrogram transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 10699–10709.

Hershey, S.; Chaudhuri, S.; Ellis, D. P.; Gemmeke, J. F.; Jansen, A.; Moore, R. C.; Plakal, M.; Platt, D.; Saurous, R. A.; Seybold, B.; et al. 2017. CNN architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, 131–135. IEEE.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141.

Huang, G.; Sun, Y.; Liu, Z.; Sedra, D.; and Weinberger, K. Q. 2016. Deep networks with stochastic depth. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, 646–661. Springer.

Huang, P.-Y.; Xu, H.; Li, J.; Baevski, A.; Auli, M.; Galuba, W.; Metze, F.; and Feichtenhofer, C. 2022. Masked autoencoders that listen. *arXiv preprint arXiv:2207.06405*.

Kong, Q.; Cao, Y.; Iqbal, T.; Wang, Y.; Wang, W.; and Plumbley, M. D. 2020. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28: 2880–2894.

Koutini, K.; Schlüter, J.; Eghbal-zadeh, H.; and Widmer, G. 2022. Efficient Training of Audio Transformers with Patchout. In *Interspeech 2022*. ISCA.

Li, X.; and Li, X. 2022. ATST: Audio representation learning with teacher-student transformer. *arXiv preprint arXiv:2204.12076*.

Li, Y.; Wu, C.-Y.; Fan, H.; Mangalam, K.; Xiong, B.; Malik, J.; and Feichtenhofer, C. 2022. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4804–4814.

Liu, X.; Lu, H.; Yuan, J.; and Li, X. 2023. CAT: Causal Audio Transformer for Audio Classification. arXiv:2303.07626.

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. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. arXiv:1711.05101.

Niizumi, D.; Takeuchi, D.; Ohishi, Y.; Harada, N.; and Kashino, K. 2021. Byol for audio: Self-supervised learning for general-purpose audio representation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Park, D. S.; Chan, W.; Zhang, Y.; Chiu, C.-C.; Zoph, B.; Cubuk, E. D.; and Le, Q. V. 2019. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *Interspeech 2019*. ISCA.

Piczak, K. J. 2015. ESC: Dataset for environmental sound classification. In *Multimedia*.

Saeed, A.; Grangier, D.; and Zeghidour, N. 2021. Contrastive learning of general-purpose audio representations. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3875–3879. IEEE.

Tokozume, Y.; Ushiku, Y.; and Harada, T. 2018. Learning from Between-class Examples for Deep Sound Recognition. arXiv:1711.10282.

Tu, Z.; Talebi, H.; Zhang, H.; Yang, F.; Milanfar, P.; Bovik, A.; and Li, Y. 2022. MaxViT: Multi-axis vision transformer. In *European Conference on Computer Vision*, 459–479. Springer.

Warden, P. 2018. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*.

Xiao, T.; Singh, M.; Mintun, E.; Darrell, T.; Dollár, P.; and Girshick, R. 2021. Early convolutions help transformers see better. *Advances in neural information processing systems*, 34: 30392–30400.

Zhang, Y.; Li, B.; Fang, H.; and Meng, Q. 2022. Spectrogram Transformers for Audio Classification. In *2022 IEEE International Conference on Imaging Systems and Techniques (IST)*, 1–6.

Zhu, W.; and Omar, M. 2023. Multiscale audio spectrogram transformer for efficient audio classification. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.