TOKEN PRUNING MEETS AUDIO: INVESTIGATING UNIQUE BEHAVIORS IN VISION TRANSFORMER-BASED AUDIO CLASSIFICATION

Taehan Lee, Woojin Lee, Hyukjun Lee
Department of Computer Science
Sogang University
Seoul, 04107, South Korea
`{alpaca,hs6113wj,hyukjunl}@sogang.ac.kr`

ABSTRACT

Vision Transformers (ViTs) have achieved state-of-the-art performance across various computer vision tasks. To reduce the high computational cost of ViTs, token pruning has been proposed to selectively remove tokens that are not crucial. While effective in vision tasks by discarding non-object regions, applying this technique to audio tasks presents unique challenges. In audio processing, distinguishing relevant from non-relevant regions is less straightforward. In this study, we applied token pruning to a ViT-based audio classification model using Mel-spectrograms and analyzed the trade-offs between model performance and computational cost. We show AudioMAE-TopK model can reduce MAC operations by 2× with less than a 1% decrease in accuracy for both speech command recognition and environmental sound classification. Notably, while many tokens from signal (high-intensity) regions were pruned, tokens from background (low-intensity) regions were frequently retained, indicating the model's reliance on these regions. In the ablation study, forcing the model to focus only on signal (high-intensity) regions led to lower accuracy, suggesting that background (low-intensity) regions contain unique, irreplaceable information for AudioMAE. In Addition, we find that when token pruning is applied, the supervised pre-trained AST model emphasizes tokens from signal regions more than AudioMAE.

1 INTRODUCTION

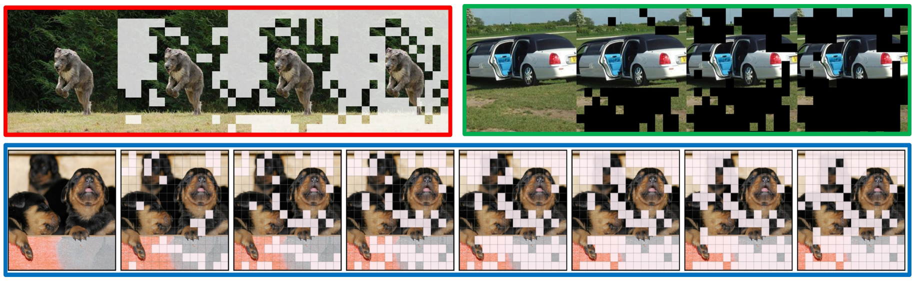


Figure 1: **Token pruning patterns with image classification tasks.** Images within red, green and blue boxes are from previous works (Rao et al., 2023; Liang et al., 2022; Fayyaz et al., 2022).

The Vision Transformer (ViT) has achieved various SOTA (state-of-the-art) results in several downstream tasks. To reduce the computational load of ViT, several token reduction methods have been proposed. In the case of image classification tasks, these methods tend to remove tokens originating from background components as shown in Fig. 1. This is a reasonable selection because such components are not necessary for classifying the object.

Several studies have shown that ViT is applicable to audio downstream tasks as well, as audio can be represented as 2D data using Mel-spectrograms. However, for audio classification tasks, it is unclear which tokens should be discarded. There are several reasons for this. For instance, background (low-intensity) regions in Mel-spectrograms do not necessarily indicate a lack of information. In the task of classifying the sound of a baby crying, losing the brief silences between cries could make it more difficult to distinguish from the sound of a siren. Furthermore, empty regions in certain frequency bands of a Mel-spectrogram can be a characteristic of the sound source, which should not be ignored.

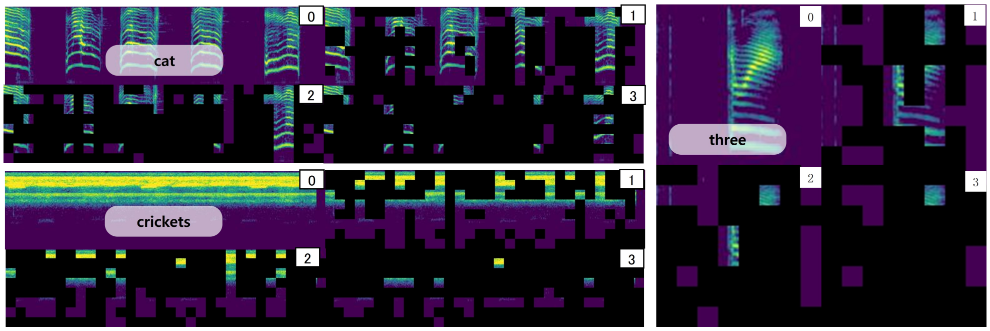


Figure 2: **Visualization of pruned tokens in AudioMAE-TopK.** The two Mel-spectrogram sets on the left are from ESC-50 and the one on the right is from SPC-2. The model's *keep-rate* was set to 0.5. For each Mel-spectrogram, the input is shown in the top-left corner (marked as '0'), and the number in each box indicates the corresponding pruning stage. More results can be found in Appendix J.

We applied token pruning to audio classification tasks and found that the pruning patterns are drastically different from image classification. Fig. 2 shows a visualization of the token pruning process using the TopK selection scheme. In this figure, we observe that both signal and background tokens are pruned as the pruning progresses. In some cases, most of the signal tokens are pruned, which leads us to ask the following questions.

- What is the role of signal versus background (noise) tokens in audio classification?
- What is the best method to select tokens for pruning, and should we preferentially retain signal or background tokens?
- Can we achieve a significant speedup in inference time by pruning a sufficient number of tokens while maintaining accuracy?
- Does the attention-based token pruning method used in vision classification tasks effectively prune the tokens?

In this study, we show an AudioMAE-TopK model, which prunes tokens according to attention scores can achieve a 2× reduction in multiply-accumulate (MAC) operations with less than a 1% drop in accuracy for speech command recognition and environmental sound classification. Through further analysis using attention scores, we confirm that the model prunes tokens with substantial confidence, not only for signal tokens but also for background tokens. In the ablation study, although signal tokens play a significant role in the classification outcome, constraining the model to retain signal tokens over selected background tokens during pruning stages leads to a non-negligible loss of accuracy, indicating the importance of their existence. In addition, we find that AST-TopK model retains more signal tokens than AudioMAE-TopK. To the best of our knowledge, this is the first work to apply token pruning to audio classification tasks using ViT while providing an in-depth analysis of the resulting pruning patterns.

2 RELATED WORKS

2.1 AUDIO SPECTROGRAM TRANSFORMERS

Since the great success of the transformer architecture (Vaswani, 2017) in Natural Language Processing (NLP), numerous works have adopted it in a variety of fields. Vision Transformer (Dosovitskiy et al., 2021) is one of the successful examples of applying the transformer to vision tasks, achieving state-of-the-art results consistently. Audio Spectrogram Transformer (AST) (Gong et al., 2021) demonstrated that the ViT is also applicable to several audio tasks by using Mel-spectrograms, which represent raw audio waveforms. Using AST, many studies have been conducted to train audio features using large amounts of unlabeled data represented as Mel-spectrograms. SS-AST (Gong et al., 2022) explored audio feature learning through a self-supervised learning method called Masked Spectrogram Patch Modeling (MSPM). AudioMAE (Huang et al., 2022) extended the masked auto-encoders (He et al., 2021) approach to audio tasks. HTS-AT (Chen et al., 2022) combined window attention and token semantic module for audio classification tasks. DFT-AT (Alex et al., 2024) introduced time-frequency decoupling techniques based on MaxViT. Recent studies such as (Niizumi et al., 2023; Li et al., 2024; Chen et al., 2024) showed that the bootstrapping method (Grill et al., 2020) can be applied to audio tasks as well.

2.2 ACCELERATING TRANSFORMERS USING TOKEN REDUCTION METHODS

ViT-based models have achieved SOTA results across several domains, but the large computational requirements hinder their deployment. Various approaches have been studied, including efficient architecture designs (Liu et al., 2021a; Fan et al., 2021), quantization (Liu et al., 2021b), knowledge distillation (Liu et al., 2024b), and early exit branches (Bakhtiarnia et al., 2021). In our work, we focus on token reduction methods, which reduce the number of tokens based on the characteristics of the input data.

A-ViT (Yin et al., 2022) proposed using a single embedding dimension as a pruning score, without modifying network architecture or adding parameters. DynamicVit and SPViT (Rao et al., 2023; Kong et al., 2022) determined which tokens to prune using prediction modules made of MLP layers. EViT (Liang et al., 2022) used the attention scores of tokens to determine which tokens carry more important information. ATS (Fayyaz et al., 2022) additionally considered the norm of the value matrix in the transformer when calculating token importance scores. METR (Liu et al., 2024a) showed integrating multi-exit layers with token pruning enables tokens originating from object regions to receive higher attention scores, making them more likely to be retained during the token reduction process. These methods extract attention matrix scores from the classification token, i.e. [CLS] token, as it gathers information from all tokens to form the final prediction. Along the token pruning, token merging have been also studied. ToMe (Bolya et al., 2023) showed that the number of tokens can be reduced through a merging strategy using the similarity between tokens without additional training. ToFu (Kim et al., 2024) suggested applying token pruning in the earlier layers and token merging in the later layers by functional linearity analysis.

2.3 EARLY EXIT METHODS ON AUDIO TASKS WITH TRANSFORMERS

HuBERT-EE (Yoon et al., 2022) introduced multiple early exit branches for speech recognition tasks, while DAISY (Lin et al., 2024) leveraged the entropy of self-supervised loss to guide early exit decisions. (Wright et al., 2024) found that training Conformer models with early exit layers from scratch yields better performance than fine-tuning a pretrained model. FastAST (Ranjan Behera et al., 2024) combined token merging and cross-model knowledge distillation in the AST framework.

3 APPLYING TOKEN PRUNING TO AUDIO CLASSIFICATION TASKS

3.1 AUDIOMAE-TOPK MODEL

To apply token pruning in audio classification tasks, we adopt AudioMAE as the backbone model due to its state-of-the-art performance across various benchmarks. We use TopK as a token pruning method since (Haurum et al., 2023) demonstrates that it's a competitive method and to easily distinguish the origin of tokens, which will be discussed in the following Section 4. After the raw audio waveform is converted into a Mel-spectrogram, it is regarded as an image so that patch embedding and positional embeddings can be applied. The token pruning module is placed between the multi-head self attention and MLP module of selected ViT blocks (the 4th, 7th, and 10th block).[1] Using all tokens in a block, the attention score for each token is calculated based on how much attention

---

[1] The choice of pruning location follows previous token pruning works (DynamicViT, EViT and METR).

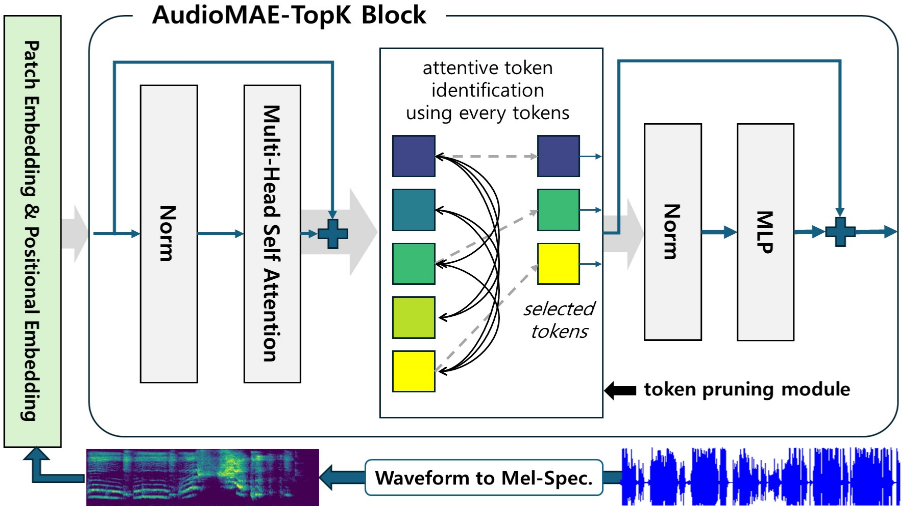

Patch Embedding & Positional Embedding
AudioMAE-TopK Block
Norm
Multi-Head Self Attention
attentive token identification using every tokens
selected tokens
Norm
MLP
token pruning module
Waveform to Mel-Spec.

Figure 3: **AudioMAE-TopK: selecting TopK attentive tokens for audio classification tasks.**

it receives from other tokens. Among N input tokens of a block, we retain only $N \times$ *keep-rate* tokens with highest scores (In Fig. 3, *keep-rate* is set to 0.6, 3 tokens are retained among the 5 input tokens). After that, the tokens not selected are pruned and only the retained tokens are passed to the next stages in the block. The same *keep-rate* is applied to all pruning-enabled blocks.

We downloaded a checkpoint of AudioMAE pre-trained on AudioSet-2M (Gemmeke et al., 2017) with a ViT-Base (ViT-B) transformer. We followed AudioMAE's configuration for pre-processing audio data: 128 Mel-spectrogram bins and $(16, 16)$ shaped patches without overlapping regions and fixed positional embeddings.

3.2 APPLYING TOKEN PRUNING TO AN AUDIO SPECTROGRAM TRANSFORMER

The Multi-Head Self-Attention (MHSA) is the key component of the Transformer (Vaswani, 2017). Tokens are transformed into three matrices: Q (Query), K (Key), and V (Value) through linear projections. The attention mechanism is applied as follows:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V} = \mathbf{A}\mathbf{V}. \tag{1}$$

$$a_{i,\text{global}}^{(b)} = \frac{1}{HN}\sum_{h=1}^{H}\sum_{n=1}^{N}\mathbf{A}[b, h, n, i] \tag{2a}$$

$$a_{i,\text{cls}}^{(b)} = \frac{1}{H}\sum_{h=1}^{H}\mathbf{A}[b, h, 0, i] \tag{2b}$$

For a batch size of B, H attention heads, and N tokens, A is a tensor called the attention map with a shape of (B, H, N, N). AudioMAE uses mean pooling for both pretraining and downstream tasks. For the input sample b, TopK method measures each token's importance using token-to-token attention score averaged over multiple heads, which is described in Eq. 2a. At each pruning enabled block of AudioMAE-TopK, tokens having the largest K attention values, $a_{i,\text{global}}^{(b)}$, are retained (not pruned) where $K = N \times$ *keep-rate*. For the ablation study in Section 5.2 (AudioMAE-TopK-CLS) and Section 5.3 (AST-TopK-CLS), we use the [CLS] token based attention scores for pruning, which is described in Eq. 2b.[2]

---

[2] The index 0 is used as the position of the [CLS] token.

3.3 EXPERIMENT SETUPS

3.3.1 DATASETS

We evaluated our approach on two audio classification datasets, measuring performance using top-1 accuracy. The Speech Commands V2 (SPC-2) (Warden, 2018) consists of 1-second recordings of 35 common speech commands, including 84,843 training samples, 9,981 validation samples, and 11,005 test samples. We fine-tuned the model using the training set and assessed its accuracy on the test data. The Environmental Sound Classification (ESC-50) (Piczak, 2015) contains 2,000 audio recordings, each 5 seconds long, across 50 classes. All results for this dataset are reported as the average of a 5-fold cross-validation, following (Gong et al., 2022). The VoxCeleb-1 (Nagrani et al., 2020) includes 150K audio samples for identification of 1,251 different speakers. We used 138,361 training samples and 8,251 testing samples. For general audio classification task, we choose balanced AudioSet task (AS-20K) consists of 21K clips and 19K evaluation clips whose lengths are 10 seconds. Before being processed by the first transformer block, the input data is converted by the patch embedding layer, producing 64 / 256 / 512 / 512 tokens for SPC-2 / ESC-50 / VoxCeleb-1 / AS-20K.

3.3.2 TRAINING HYPERPARAMETERS

Unlike AudioMAE, we did not use any masking strategy during fine-tuning, including SpecAug (Park et al., 2019), because training with reduced tokens already imposes a large amount of masking. We found that using layer-wise learning rate decay (Bao et al., 2022) helps ESC-50 perform better in token pruning. We used a higher minimum learning rate of 10^{-5} except for VoxCeleb-1. We use the different settings for starting epoch for shrinking and the shrinking period for each dataset. We used padding with minimum values for the samples that were too short to fill the target time. Other configurations follow AudioMAE and detailed numbers are listed in Appendix A. We trained the model with mixed precision (Micikevicius et al., 2018) and evaluated using float32.

3.4 BENCHMARK RESULTS AND VISUALIZATION OF PRUNED TOKENS

Keep rate (kr)	**SPC-2**		**ESC-50**		**VoxCeleb-1**		**AS-20K**	
	Top-1 Acc (%)	**MACs (G)**	**Top-1 Acc (%)**	**MACs (G)**	**Top-1 Acc (%)**	**MACs (G)**	**mAP -**	**MACs (G)**
1.0	97.95	5.61	94.30	23.11	94.26	48.57	0.372	48.57
0.9	97.79	4.93	93.90	20.02	94.28	41.79	0.366	41.79
0.8	97.67	4.30	93.55	17.29	94.24	36.04	0.362	36.04
0.7	97.64	3.72	93.50	15.02	93.90	31.12	0.357	31.12
0.6	97.59	3.27	93.75	13.05	92.87	27.12	0.352	27.12
0.5	97.28	2.81	93.40	11.37	91.32	23.65	0.344	23.65

Table 1: **Results of AudioMAE-TopK.** torchprofile (Liu, 2021) was used to measure Multiply-Accumulate Computations (MACs). Comparing to AudioMAE, our baseline accuracy without token pruning was slightly higher on ESC-50 (+0.2%), AS-20K (+0.2%) and lower on SPC-2 (-0.3%), VoxCeleb-1 (-0.5%). Throughput comparison is shown in Appendix B.

As demonstrated in Tab. 1, token pruning by AudioMAE-TopK shows tradeoff relationship between reduction of MACs and drop of accuracy compared to the baseline (*keep-rate*=1.0). Specifically, MACs are reduced by roughly a factor of two, while accuracy decreases by 0.67% and 0.90% for SPC-2 and ESC-50, respectively, when using a *keep-rate* of 0.5.

As shown in Fig. 1, the token reduction methods for image classification tasks (Yin et al., 2022; Rao et al., 2023; Liang et al., 2022; Yin et al., 2022; Fayyaz et al., 2022; Liu et al., 2024a) tend to retain tokens relevant to the label while discarding background components. However, it is unclear which types of patches (tokens) are most important for audio classification. Initially, we expected that tokens representing signal (high-intensity) patches would receive higher attention and thus to be mostly retained. Contrary to our expectations, the visualization results indicate that background tokens often receive higher attention scores than signal tokens during pruning stages. In Mel-spectrograms of the SPC-2 dataset, we can easily distinguish between tokens representing signals and those representing noise/background, as each sample contains only a single word. Interestingly, the right box in Fig. 2 shows that most tokens covering the voice region were discarded. As shown in other datasets (Appendix J) the trend of token pruning patterns in audio classification tasks differs from that in image tasks, prompting us to investigate the effects of pruning on different types of tokens.

4 Analysis of Token Pruning in AudioMAE-TopK

4.1 Signal vs. Background Tokens

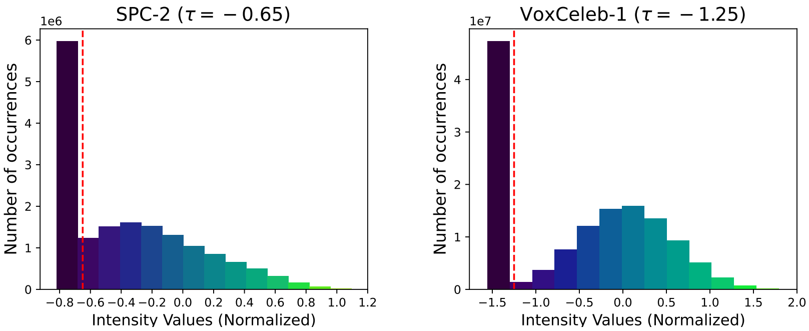


Figure 4: **Histogram of normalized Mel-spectrogram intensity values obtained from 1024 samples for each dataset.** The histogram reveals that high-intensity values (yellow regions in the Mel-spectrogram) are infrequent and not visible due to their small proportion in the dataset. We pick the threshold value, τ, for each dataset from the 2nd bin of each histogram. For SPC-2, ESC-50, VoxCeleb-1 and AS-20K, τ is set to -0.65, -0.65, -1.25 and -1.0. Signal areas account for 63% in SPC-2, 81% in ESC-50, 65% in VoxCeleb-1, and 90% in AS-20K, with the rest classified as background. Histrogram for other datasets are in Appendix C.

We begin our analysis of the visualization results by quantifying how many signal tokens are retained during the pruning process. We define threshold values, τ, for each dataset to distinguish background/noise/pad tokens from signal tokens. Based on the observations from Fig. 4, we classify tokens as originating from signal areas if the corresponding patch's mean intensity value is greater than τ; Otherwise, they are considered as background tokens. Let N represent the total number of input tokens, and R the set of retained tokens. A set of tokens retained from signal areas is denoted as R_s, while one for background areas is denoted as R_b, such that $R = R_s \cup R_b$. Similarly, we denote the sets for the pruned tokens as $P = P_s \cup P_b$. Superscripts are used to indicate tokens at different pruning stages; for example, R_s^2 refers to the set of signal tokens retained after the second pruning stage. This classification allows us to analyze the impact of different token types based on their retention status throughout the pruning stages.

Keep rate	**SPC-2**			**ESC-50**			**VoxCeleb-1**			**AS-20K**		
	0.9	**0.7**	**0.5**	**0.9**	**0.7**	**0.5**	**0.9**	**0.7**	**0.5**	**0.9**	**0.7**	**0.5**
loc 1/A	60.0	46.8	33.1	74.9	58.4	41.6	60.0	46.7	33.3	83.5	65.0	46.4
loc 2/A	60.5	47.1	33.0	74.8	58.2	41.5	60.0	46.7	33.3	83.5	65.1	46.3
loc 3/A	60.0	47.6	33.1	75.1	58.6	41.5	60.0	46.8	33.4	83.6	65.1	46.3
loc 1/B	90.7	70.4	50.0	90.3	70.5	50.2	90.1	70.1	50.0	90.0	70.1	50.0
loc 2/B	82.8	50.0	25.0	81.3	49.3	25.1	81.1	49.2	25.0	81.0	49.2	25.0
loc 3/B	75.0	36.0	12.5	73.6	34.8	12.5	73.1	34.6	12.5	73.0	34.5	12.5

Table 2: **Percentage (%) of retained signal tokens to available tokens at each pruning stage (A) and to the total number of signal tokens in the input (B) with varying *keep-rates*.** Loc 1,2, and 3 represent the 4th, 7th, and 10th attention blocks respectively.

We measured two different ratios for retained signal tokens. Ratio **A** is defined as $|R_s^i|/(|R^{i-1}| + |P^{i-1}|)$, which denotes the number of retained signal tokens at stage i divided by the number of all

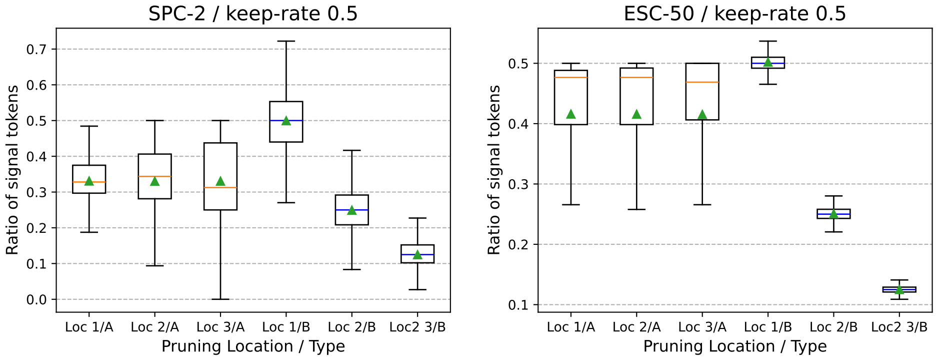


Figure 5: **Distribution of signal token ratios retained during token pruning.** While the average ratio of retained signal tokens to total candidate tokens (ratio **A**) remains similar across different pruning stages, the variance is high across the dataset. More results are in Appendix F.

tokens before pruning, except for the first location where the ratio is $|R_s^1|/|N|$. Ratio **B** is defined as $|R_s^i|/|N_s|$, which measures the ratio of retained signal tokens at stage i to the total number of signal tokens in the input Mel-spectrogram. Interestingly, Tab. 2 shows that the portion of signals in retained tokens with respect to the number of signals in input (ratio **B**) decreases proportionally to the *keep-rate*.

4.2 COMPARING THE ATTENTION SCORE OF RETAINED AND PRUNED TOKENS

Keep rate	**SPC-2**			**ESC-50**			**VoxCeleb-1**			**AS-20K**		
	0.9	**0.7**	**0.5**	**0.9**	**0.7**	**0.5**	**0.9**	**0.7**	**0.5**	**0.9**	**0.7**	**0.5**
$\Gamma(G_1, R_s)$	1.31	1.12	1.11	1.15	1.11	1.11	1.38	1.27	1.22	1.17	1.13	1.13
$\Gamma(G_2, R_s)$	5.29	1.82	1.47	1.84	1.58	1.45	2.86	1.96	1.42	1.60	1.41	1.35
$\Gamma(G_3, R_s)$	13.03	4.10	2.17	2.60	1.86	1.73	11.31	5.51	3.03	2.58	2.04	1.88
$\Gamma(G_1, R_b)$	1.30	1.11	1.11	1.13	1.12	1.10	1.38	1.30	1.28	1.12	1.09	1.11
$\Gamma(G_2, R_b)$	5.28	1.82	1.46	1.75	1.51	1.52	2.89	2.14	1.52	1.52	1.33	1.20
$\Gamma(G_3, R_b)$	13.14	4.05	2.13	2.54	2.04	1.83	10.99	5.40	3.20	2.00	1.60	1.98

Table 3: **Ratio of the attention scores of retained and pruned tokens for signals and backgrounds, averaged over groups.**

We compare the attention scores of the retained and pruned tokens because AudioMAE-TopK uses the scores to prune tokens. For analysis, we group attention blocks with respect to the pruning locations: $G_1 = (1, 2, 3, 4)$ / $G_2 = (5, 6, 7)$ / $G_3 = (8, 9, 10)$ and split tokens at each group into two sets for retained and pruned tokens. For block l, R_s^l, R_b^l and P^l denotes the set of retained signal, retained background, and pruned tokens at their corresponding pruning block (i.e. $R_s^1 = R_s^2 = R_s^3 = R_s^4$). We define $\gamma(l, R_s^l)$ as *attn_score*(R_s^l) / *attn_score*(P^l), representing the ratio of retained signal tokens' average attention score to pruned tokens' average attention score at block l. Similarly, $\gamma(l, R_b^l)$ is *attn_score*(R_b^l) / *attn_score*(P^l) at block l. Using this ratio, we wanted to keep track of attention values for retained tokens over pruned tokens at each block. $\Gamma(G_i, R_s)$ is defined as $\frac{1}{|G_i|}\sum_j \gamma(G_i[j], R_s^{G_i[j]})$, representing the averaged ratio value across the blocks in the same group. Tab. 3 presents the average ratio values over groups of blocks, and Fig. 6 illustrates the trend of ratios at the block level, rather than the group level.

In Fig. 6, retained tokens show much larger attention scores than pruned ones as tokens are processed, regardless of their origins, as Γ is greater than 1. The drop of attention score ratio right after 1st and 2nd pruning location indicates that after the tokens are pruned, the model is relatively unsure which tokens need to be pruned among retained ones before computing next level of attention values. Tab. 3 shows model becomes more confident about its token selection for pruning when the *keep-rate* increases.

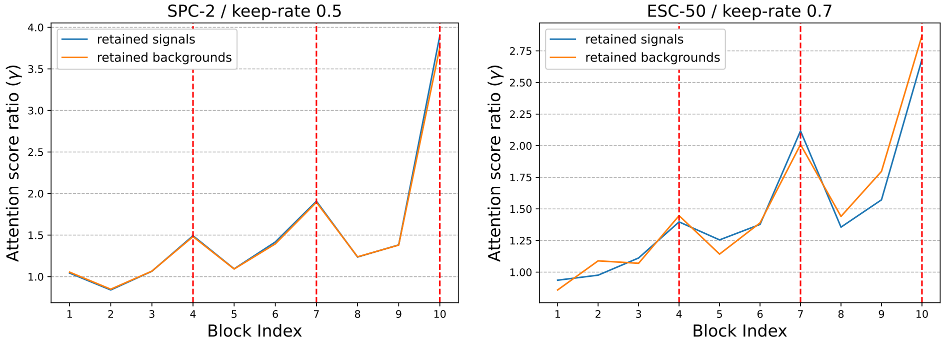


Figure 6: **Ratio of the attention scores of retained and pruned tokens for signals and backgrounds.** In the left plot (SPC-2), the two graphs nearly overlap because of the similarity in the ratio values. Red dashed lines indicate the locations of token pruning at the 4th, 7th, and 10th blocks. More results can be found in Appendix D.

4.3 ANALYZING THE PRUNING OF SIGNAL TOKENS

Each ViT block includes a residual connection over the self-attention layer. One of the mysterious behaviors in pruning was the pruning of signal tokens. To understand this, we examined the effect of the residual connection on the output of the multi-headed self-attention block as pruning progresses by comparing their L_2-norms.

We measured the L_2-norm ratio, $\mathcal{R}(x) = ||x||_2/||self\_attention(x)||_2$, where x represents a tensor for the residual path from the output of the previous block as depicted in Fig. 3. In Fig. 7, the norm ratios are reported. Interestingly, they become smaller in the later blocks, indicating that the self-attention layer is giving more influence to the model's classification result. This trend was consistent regardless of pruning level. Our interpretation of this behavior is as follows. Initially, many tokens merge their information gradually into representative signal and background tokens. Then, only a small number of representative signal and background tokens are retained throughout the pruning process, and they are highly attentive with each other in the later blocks to perform classification, which explains why many signal tokens can be pruned without a loss of accuracy.

In contrast to this result, ViT in image classification tasks observe an opposite trend to our findings (Raghu et al., 2021). In image classification tasks, the L_2-norm ratio increases in the later blocks. That is because final objects identified in the later layers are not necessarily related (attentive) to background to perform the image classification task.

We also observed that the norm ratio normalized to the model without pruning (*keep-rate*=1.0) becomes smaller if we prune more tokens by setting a lower *keep-rate*, as illustrated in the right plot of Fig. 7.

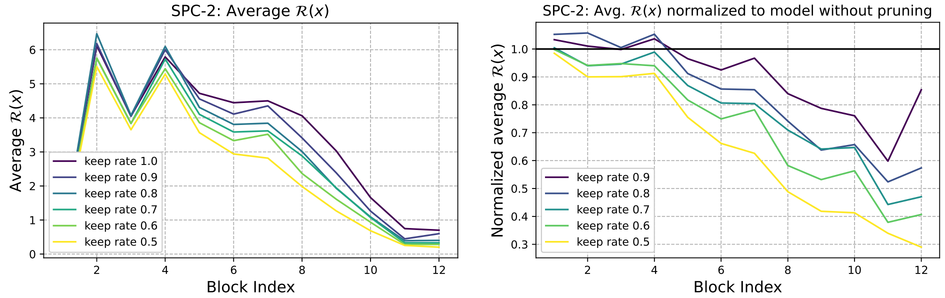


Figure 7: **Ratios of the residual connection norm to the self-attention norm (left) and the same ratios normalized to the model without token pruning (right).** More results can be found in Appendix E.

5 ABLATION STUDY

5.1 SWAPPING TOKENS IN DIFFERENT GROUPS

To find out how much signal and background tokens contribute to the classification accuracy, we conducted three perturbation tests over the set of retained tokens. For the token selection, we enforce the following rules. Let X and Y be the list of tokens sorted in decreasing order by their attention scores given by Eq. 2a. The swapping algorithm is described in Algorithm 1. We denote R_s, R_b, P_s, P_b as sorted lists of retained signal/background and pruned signal/background tokens respectively. In test **I** and test **II**, we perform *Perturb*(R_s, P_s) and *Perturb*(R_b, P_b) for each pruning location to measure the impact of swapping retained and pruned tokens of the same category. In test **III**, we perform *Perturb*(R_b, P_s) to see whether forcing the usage of pruned signal tokens instead of retained background tokens can improve the model's prediction. The results are reported in Tab. 4.

Algorithm 1 Swapping Tokens in Different Groups

```
function PERTURB(X, Y)
    if len($X$) $\leq$ len($Y$) then
        $X \leftarrow Y[:\text{len}(X)]$
    else
        $n \leftarrow \text{len}(X) - \text{len}(Y)$
        $X \leftarrow \text{concat}(X[:n], Y)$
    end if
end function
```

Keep rate	**SPC-2**					**ESC-50**				
	0.9	**0.8**	**0.7**	**0.6**	**0.5**	**0.9**	**0.8**	**0.7**	**0.6**	**0.5**
baseline	97.79	97.67	97.64	97.59	97.28	93.90	93.55	93.50	93.75	93.40
Test **I**	97.74	97.33	95.87	89.47	83.30	93.40	92.90	93.10	91.30	76.40
Test **II**	97.69	97.14	96.78	95.85	94.76	93.85	93.45	93.50	92.90	90.95
Test **III**	97.27	93.88	92.98	92.60	92.66	93.35	93.20	92.85	92.50	91.85

Table 4: **Top-1 accuracy results from perturbation tests.**

In test **I** and **II**, accuracy drops show that R_s carry more information than R_b related to the final prediction. With the lower *keep-rates*, the role of R_s becomes more important. Nevertheless, we cannot ignore the impact of R_b because the accuracy drops more than about 2% when it is swapped with P_b. Interestingly, test **III** does not show improvement over the baseline model. In addition, except for one case with *keep-rate*=0.5 in ESC-50, test **III** reported lower scores than test **II**. This implies that background tokens carry distinct and irreplaceable information to the signal tokens. This explains why AudioMAE-TopK model maintains accuracy even when retaining a high ratio of background tokens in the low *keep-rate* case.

5.2 USING [CLS] TOKEN FOR TOKEN SELECTION

In AudioMAE-TopK, we used the attention score based on Eq. 2a whereas EViT, ATS, and METR used attention scores with respect to the [CLS] token as in Eq. 2b. To investigate whether retaining a large number of background tokens is attributable to token-to-token attention scores and the use of mean pooling for prediction, we observed pruning behavior of AudioMAE-TopK-CLS, which prunes using the [CLS] based attention score. For this test, we used the same hyper-parameters as training AudioMAE-TopK. As depicted in Fig. 12 and Fig. 13, the pruning pattern exhibits similar behavior when compared to that of the AudioMAE-TopK, retaining both signal and background tokens. We also find that the accuracy dropped more in ESC-50 task (Tab. 5, MAE).

5.3 USING SUPERVISED PRETRAINED MODEL (AST)

We also compare the pruning patterns of AudioMAE-TopK with those of AST-TopK, which is pre-trained using supervised learning. We downloaded a checkpoint of AST-CLS (ViT-B, stride=16, 0.442 mAP on AS-2M) which uses a [CLS] token and pretrain AST-MP on AudioSet-2M using mean pooling for prediction, following the same recipe in (Gong et al., 2021). AST-MP records 0.435 mAP on AS-2M. AST-TopK gives higher attention scores to signal tokens than to background tokens (Appendix I), and its pruning patterns are also visually distinguishable from those of AudioMAE-TopK, whether a CLS token or mean pooling is used (Appendix K). AST-TopK-CLS retains accuracy better than AudioMAE-TopK on ESC-50 (Tab. 5, AST).

Keep rate	**SPC-2 (MAE)**			**ESC-50 (MAE)**			**SPC-2 (AST)**			**ESC-50 (AST)**		
	0.9	**0.7**	**0.5**	**0.9**	**0.7**	**0.5**	**0.9**	**0.7**	**0.5**	**0.9**	**0.7**	**0.5**
Top-1	97.7	97.5	97.1	90.2	88.0	83.3	97.2	97.0	97.0	94.7	94.9	94.5
loc 1/A	60.0	46.5	33.1	74.9	58.3	41.4	65.2	57.9	47.2	75.9	60.6	44.3
loc 2/A	60.1	47.1	33.2	74.7	58.0	41.4	67.8	65.0	48.7	76.6	62.4	46.9
loc 3/A	60.0	47.6	33.2	75.1	58.6	41.4	72.6	67.7	49.3	76.5	62.7	47.0
loc 1/B	90.6	70.3	50.0	90.3	70.0	49.9	99.0	88.6	73.3	92.5	76.3	58.4
loc 2/B	82.9	50.0	25.0	81.2	49.0	24.8	93.5	70.7	38.3	85.3	57.9	33.7
loc 3/B	75.0	35.9	12.5	73.4	34.6	12.4	91.6	52.7	19.6	76.5	41.0	17.5

Table 5: **Percentage(%) of retained signal tokens to available tokens at each pruning stage (A) and to the total number of signal tokens in the input (B) with varying *keep-rates* of AudioMAE-TopK-CLS and AST-TopK-CLS.** AST-CLS / AST-MP achieved accuracies of 97.1 / 97.1 on SPC-2 and 95.0 / 94.5 on ESC-50 without pruning. For AST-TopK-MP with keep rates of 0.9, 0.7, and 0.5, the model achieved accuracies of 94.5, 94.3, and 93.5 on ESC-50, and 97.0, 97.1, and 97.0 on SPC-2, respectively.

6 CONCLUSION

In conclusion, this study shows that the attention-based token pruning method can be successfully applied to audio tasks, reducing computational costs by 2× MACs with less than a 1% accuracy drop in both speech command recognition and environmental sound classification. We find that token pruning patterns differ from those in image classification tasks and can vary based on the pretrained model. AudioMAE distinguishes retained tokens by assigning them larger attention scores compared to pruned ones, regardless of whether they originate from signal or background regions. Additionally, AST assigns higher attention scores to signal tokens than to background tokens. We show that for AudioMAE, while signal tokens play a more prominent role in the model's predictions, background tokens also provide essential and irreplaceable information that contributes to the overall model performance.

7 LIMITATIONS AND DISCUSSIONS

Due to limited computational resources, our study focuses on the ViT-B configuration. **Pruning patterns may vary with different model capacities and pretraining methods**. **Other pruning methods** such as DynamicViT, which uses MLP modules as token score predictor, **may exhibit different pruning patterns**. Expanding our experiments to **a wider range of audio classification tasks** like emotion recognition could offer deeper insights, as each task has unique characteristics. AudioMAE shows more accuracy drop than AST when token pruning is applied but still maintains competitive performance by retaining lots of background tokens. This difference can be attributed to their pretraining objectives: In AudioMAE, the model processes audio by considering not only signal but also background tokens, as all patches are masked during training. In contrast, AST focuses primarily on signal tokens with higher intensity values. This distinction can lead to different pruning behaviors: AudioMAE-TopK pruning retains both signal and background tokens for classification tasks, while AST-TopK pruning retains more signal tokens. We also believe that there are opportunities to develop **token pruning methods tailored for specific audio tasks**, such as speech recognition, using domain knowledge.

REFERENCES

Tony Alex, Sara Ahmed, Armin Mustafa, Muhammad Awais, and Philip JB Jackson. Dtf-at: Decoupled time-frequency audio transformer for event classification. In *AAAI*, 2024.

Arian Bakhtiarnia, Qi Zhang, and Alexandros Iosifidis. Multi-exit vision transformer for dynamic inference, 2021.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEit: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations*, 2023.

Ke Chen, Xingjian Du, Bilei Zhu, Zejun Ma, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022.

Wenxi Chen, Yuzhe Liang, Ziyang Ma, Zhisheng Zheng, and Xie Chen. Eat: Self-supervised pre-training with efficient audio transformer. *arXiv preprint arXiv:2401.03497*, 2024.

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*, 2021.

Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6824–6835, 2021.

Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive token sampling for efficient vision transformers. In *European Conference on Computer Vision*, 2022.

Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 776–780. IEEE, 2017.

Yuan Gong, Yu-An Chung, and James Glass. AST: Audio Spectrogram Transformer. In *Proc. Interspeech 2021*, pp. 571–575, 2021. doi: 10.21437/Interspeech.2021-698.

Yuan Gong, Cheng-I Lai, Yu-An Chung, and James Glass. Ssast: Self-supervised audio spectrogram transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Joakim Bruslund Haurum, Sergio Escalera, Graham W. Taylor, and Thomas Baltzer Moeslund. Which tokens to use? investigating token reduction in vision transformers. *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 773–783, 2023.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv:2111.06377*, 2021.

Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer. Masked autoencoders that listen. *Advances in Neural Information Processing Systems*, 35:28708–28720, 2022.

Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Token fusion: Bridging the gap between token pruning and token merging. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1383–1392, 2024.

Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, et al. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XI*, pp. 620–640. Springer, 2022.

Xian Li, Nian Shao, and Xiaofei Li. Self-supervised audio teacher-student transformer for both clip-level and frame-level tasks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022.

Tzu-Quan Lin, Hung-yi Lee, and Hao Tang. Daisy: Data adaptive self-supervised early exit for speech representation models. *arXiv preprint arXiv:2406.05464*, 2024.

Dongyang Liu, Meina Kan, Shiguang Shan, and CHEN Xilin. A simple romance between multi-exit vision transformer and token reduction. In *The Twelfth International Conference on Learning Representations*, 2024a.

Ruiping Liu, Kailun Yang, Alina Roitberg, Jiaming Zhang, Kunyu Peng, Huayao Liu, Yaonan Wang, and Rainer Stiefelhagen. Transkd: Transformer knowledge distillation for efficient semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–17, 2024b.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021a.

Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021b.

Zhijian Liu. torchprofile. `https://github.com/zhijian-liu/torchprofile`, 2021. Accessed: 2024-09-23.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training. In *International Conference on Learning Representations*, 2018.

Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. Voxceleb: Large-scale speaker verification in the wild. *Computer Speech & Language*, 60:101027, 2020.

Daisuke Niizumi, Daiki Takeuchi, Yasunori Ohishi, Noboru Harada, and Kunio Kashino. BYOL for Audio: Exploring pre-trained general-purpose audio representations. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:137–151, 2023.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.

Karol J Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 1015–1018, 2015.

Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in neural information processing systems*, 34:12116–12128, 2021.

Swarup Ranjan Behera, Abhishek Dhiman, Karthik Gowda, and Aalekhya Satya Narayani. Fastast: Accelerating audio spectrogram transformer via token merging and cross-model knowledge distillation. *arXiv e-prints*, pp. arXiv–2406, 2024.

Yongming Rao, Zuyan Liu, Wenliang Zhao, Jie Zhou, and Jiwen Lu. Dynamic spatial sparsification for efficient vision transformers and convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):10883–10897, 2023.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

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

George August Wright, Umberto Cappellazzo, Salah Zaiem, Desh Raj, Lucas Ondel Yang, Daniele Falavigna, Mohamed Nabih Ali, and Alessio Brutti. Training early-exit architectures for automatic speech recognition: Fine-tuning pre-trained models or training from scratch. In *2024 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, pp. 685–689. IEEE, 2024.

Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10809–10818, 2022.

Ji Won Yoon, Beom Jun Woo, and Nam Soo Kim. Hubert-ee: Early exiting hubert for efficient speech recognition. *arXiv preprint arXiv:2204.06328*, 2022.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

A HYPERPARAMETER SETTINGS

Hyperparameters	**SPC-2**	**ESC-50**	**VoxCeleb-1**	**AS-20K**
Optimizer	AdamW (Loshchilov & Hutter, 2019)			
Optimizer momentum	$\beta_1 = 0.9,\ \beta_2 = 0.95$			
Layer decay (Bao et al., 2022)	0.75			
Weight decay	0.0001			
Learning rate schedule (lr_{base})	Cosine (Loshchilov & Hutter, 2017)			
Base learning rate	0.001			
Minimum learning rate	0.00001		0.000001	0.00001
Warm-up epochs	4			
Total epochs	90 / 30*	120 / 30*	60	60
Shrink start epoch	10 / 5*	20 / 5*	10	30
Shrink epochs (Liang et al., 2022)	30 / 15*	40 / 15*	20	20
Batch Size	256	64	32	16
GPUs	2			
Drop path	0.1			
Mixup (Zhang et al., 2018)	0.5	0.0	0.0	0.5
Loss Function	BCE	CE	CE	BCE
Dataset Mean for Normalization	-6.846	-6.627	-6.370	-4.268
Dataset Std for Normalization	5.565	5.359	3.074	4.569

Table 6: **Hyperparameter configurations of AudioMAE-TopK.** Following AudioMAE, the effective learning rate lr_{eff} is $lr_{\text{base}} \times \frac{\text{batch\_size}}{256}$. For AS-20K task, we applied AudioMAE recipe until shrink start epoch since the model suffers large mAP drop without masking augmentations. Our implementation is primarily based on both AudioMAE (Huang et al., 2022) and EViT (Liang et al., 2022). We used two RTX 4090 for fine-tuning. * For fine-tuning on AST, we only add those parameters for applying token pruning and the rest follows the original paper (Gong et al., 2021).

B THROUGHPUT MEASUREMENTS OF AUDIOMAE-TOPK

Keep rate	SPC-2		ESC-50		VoxCeleb-1		AS-20K	
	Top-1 (%)	Throughput (sample/s)	Top-1 (%)	Throughput (sample/s)	Top-1 (%)	Throughput (sample/s)	mAP -	Throughput (sample/s)
1.0	97.95	2625	94.30	639	94.26	270	0.371	269
0.9	97.79	2982	93.90	740	94.28	314	0.366	317
0.8	97.67	3307	93.55	851	94.24	370	0.362	359
0.7	97.64	3702	93.50	970	93.90	431	0.357	428
0.6	97.59	4251	93.75	1112	92.87	489	0.352	493
0.5	97.28	4650	93.40	1247	91.32	554	0.344	544

Table 7: **Throughput measurements of AudioMAE-TopK.** Throughput was measured using a single RTX 4090 with a batch size of 128 (SPC-2 & ESC-50) and 16 (VoxCeleb-1 & AS-20K), averaged over 50 runs. The number of tokens of VoxCeleb-1 and AS-20K tasks are the same.

C INTENSITY HISTOGRAM OF ESC-50 & AUDIOSET-20K DATASET

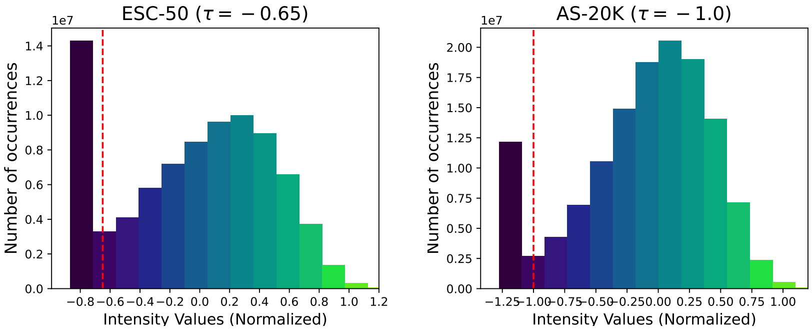


Figure 8: **Histogram of normalized Mel-spectrogram values obtained from 1024 samples.**

D MORE ATTENTION SCORE GRAPHS OF AUDIOMAE-TOPK

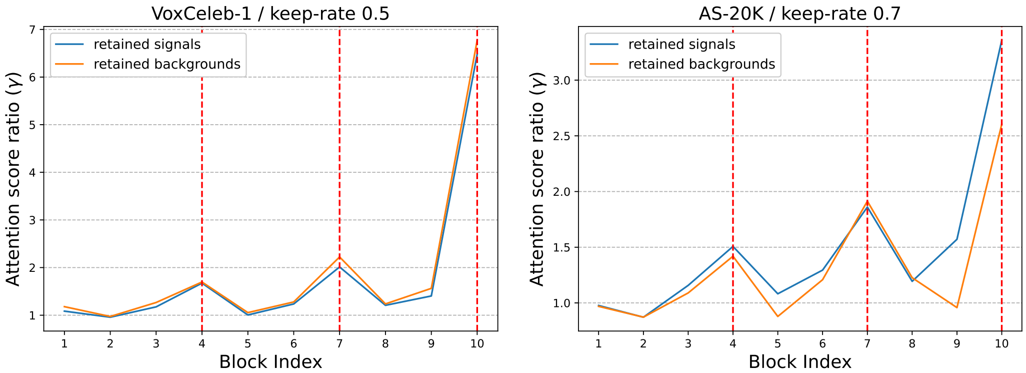


Figure 9: **Ratio of the attention scores of retained and pruned tokens for signals and backgrounds (AudioMAE-TopK)**.

E MORE ATTENTION NORM RATIO GRAPHS OF AUDIOMAE-TOPK

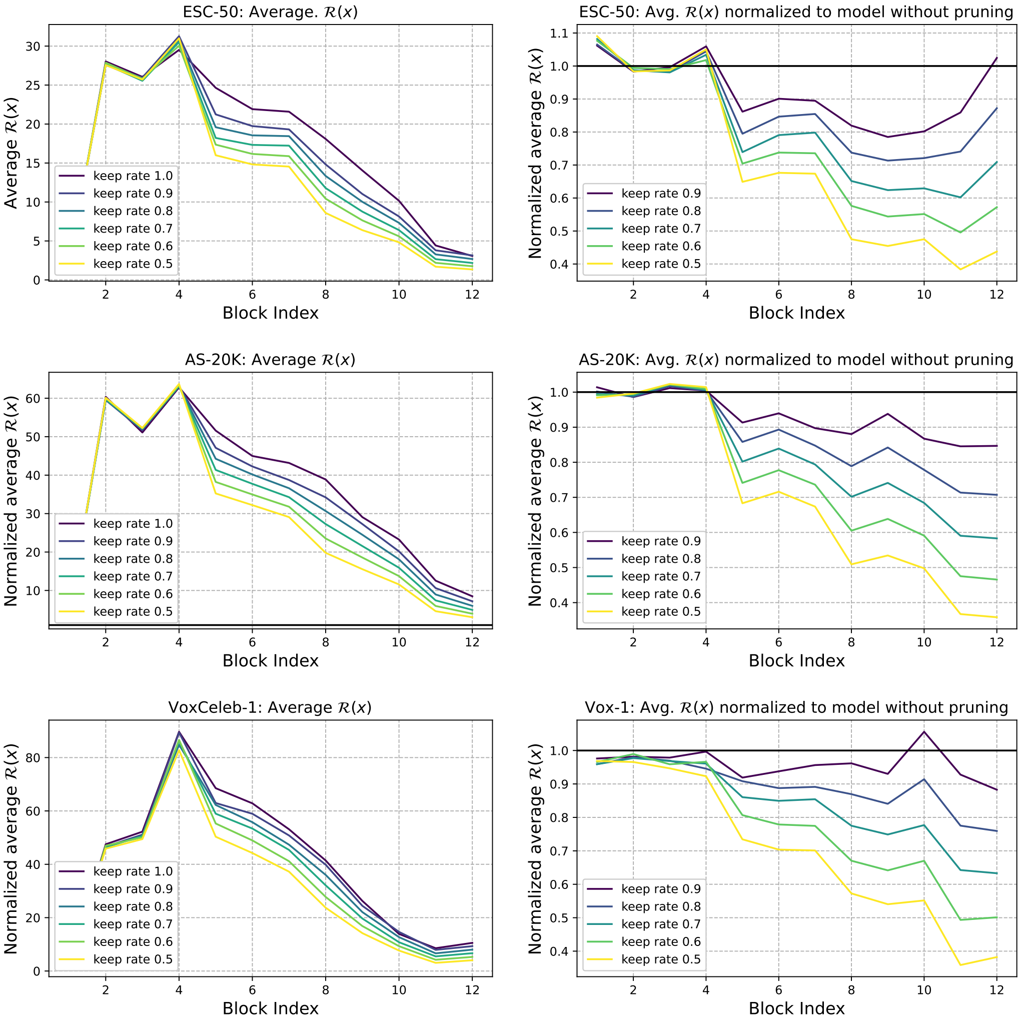


Figure 10: **Ratio of residual connections' norm to the self-attention norm (left) and normalized ratios compared to the one without token pruning (right).**

F SIGNAL TOKEN RATIO FIGURES OF AUDIOMAE-TOPK

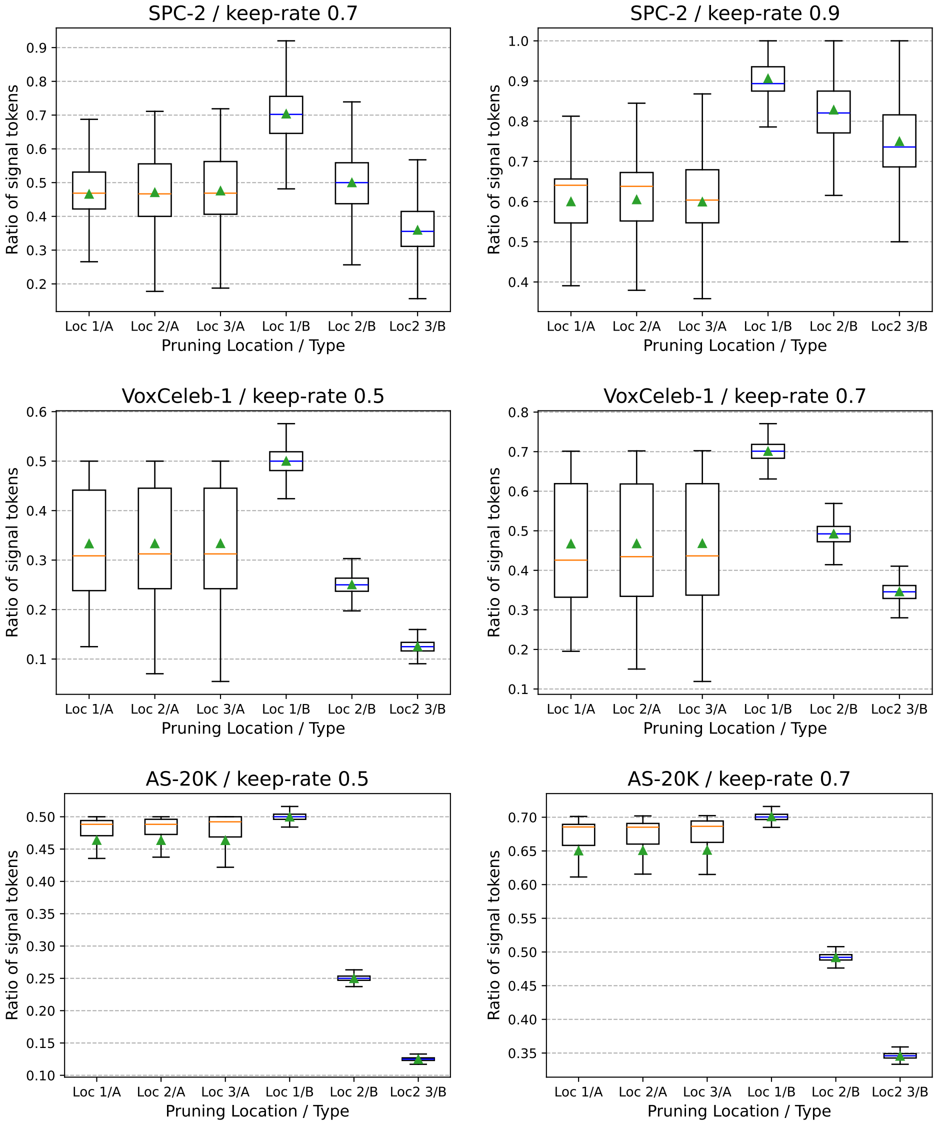


Figure 11: **Distribution of signal token ratios retained during token pruning.**

G MORE FIGURES OF AUDIOMAE-TOPK-CLS

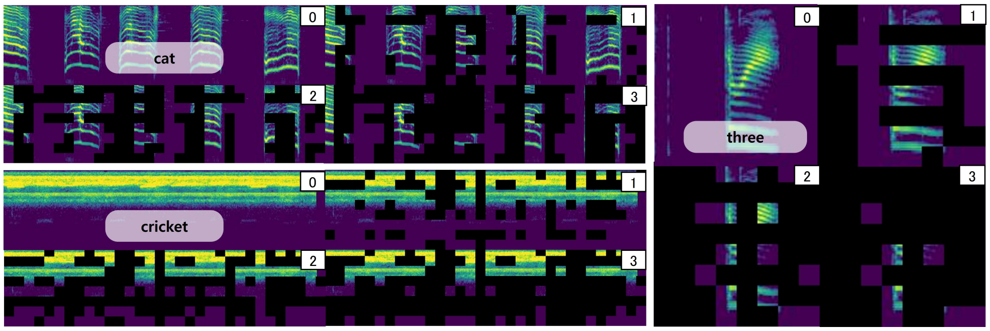


Figure 12: **Visualization of pruned tokens in AudioMAE-TopK-CLS.** The chosen samples are the same as in Fig. 2. For two ESC-50 samples (left), *keep-rates* are set to 0.7 as with a lower *keep-rate* the model suffers significant accuracy loss and thus its results may be unusable. *keep-rate* is set to 0.5 for a sample from SPC-2 (right).

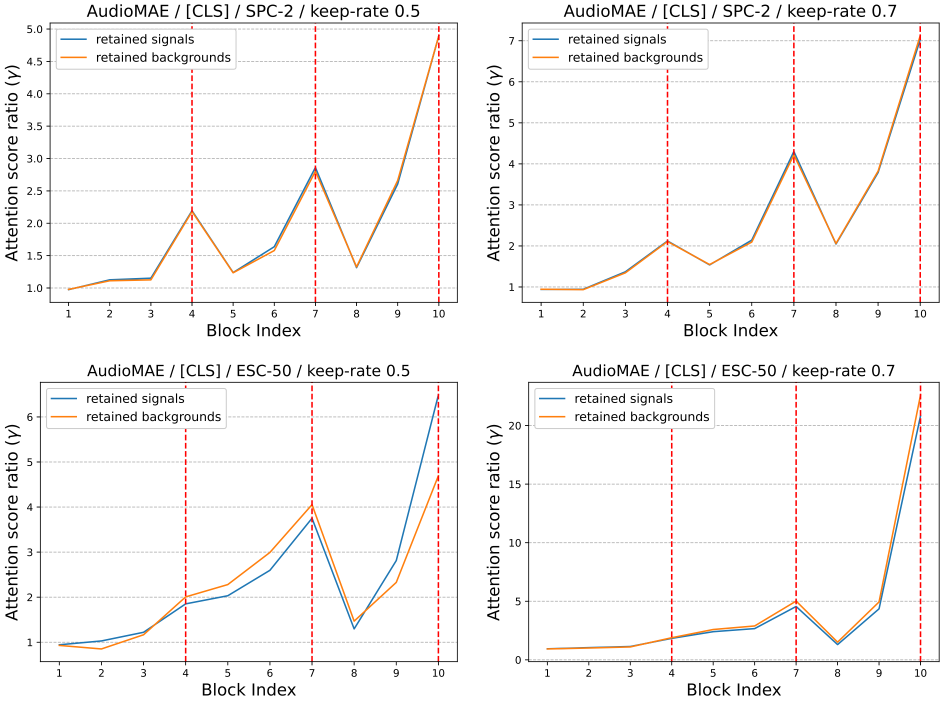
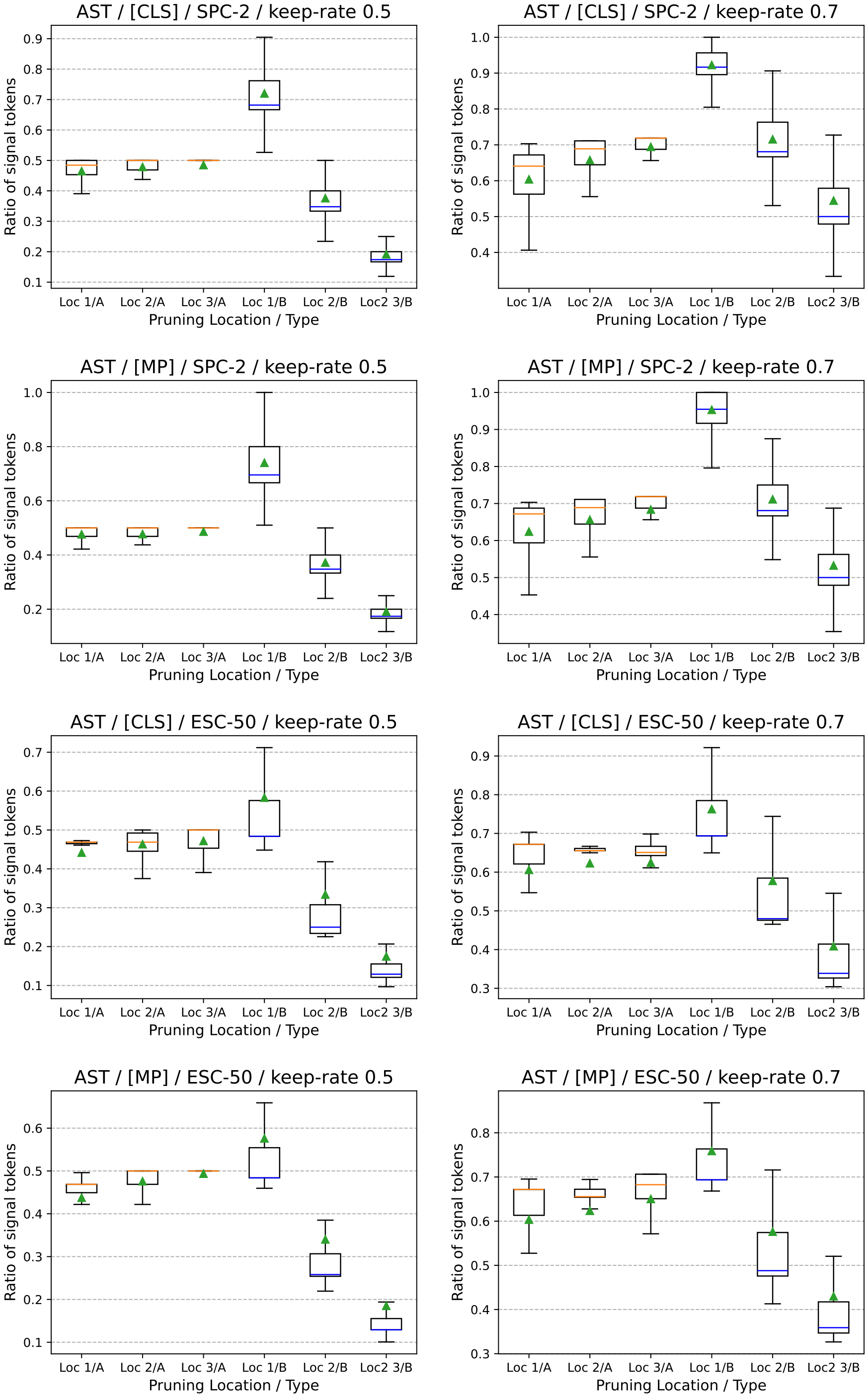


Figure 13: **Ratio of the attention scores of retained and pruned tokens for signals and backgrounds (AudioMAE-TopK-CLS)**.

H SIGNAL TOKEN RATIO FIGURES OF AST


AST / [CLS] / SPC-2 / keep-rate 0.5

AST / [CLS] / SPC-2 / keep-rate 0.7

AST / [MP] / SPC-2 / keep-rate 0.5

AST / [MP] / SPC-2 / keep-rate 0.7

AST / [CLS] / ESC-50 / keep-rate 0.5

AST / [CLS] / ESC-50 / keep-rate 0.7

AST / [MP] / ESC-50 / keep-rate 0.5

AST / [MP] / ESC-50 / keep-rate 0.7

Figure 14: **Distribution of signal token ratios retained during token pruning (AST-TopK).**

I ATTENTION SCORE GRAPHS OF AST

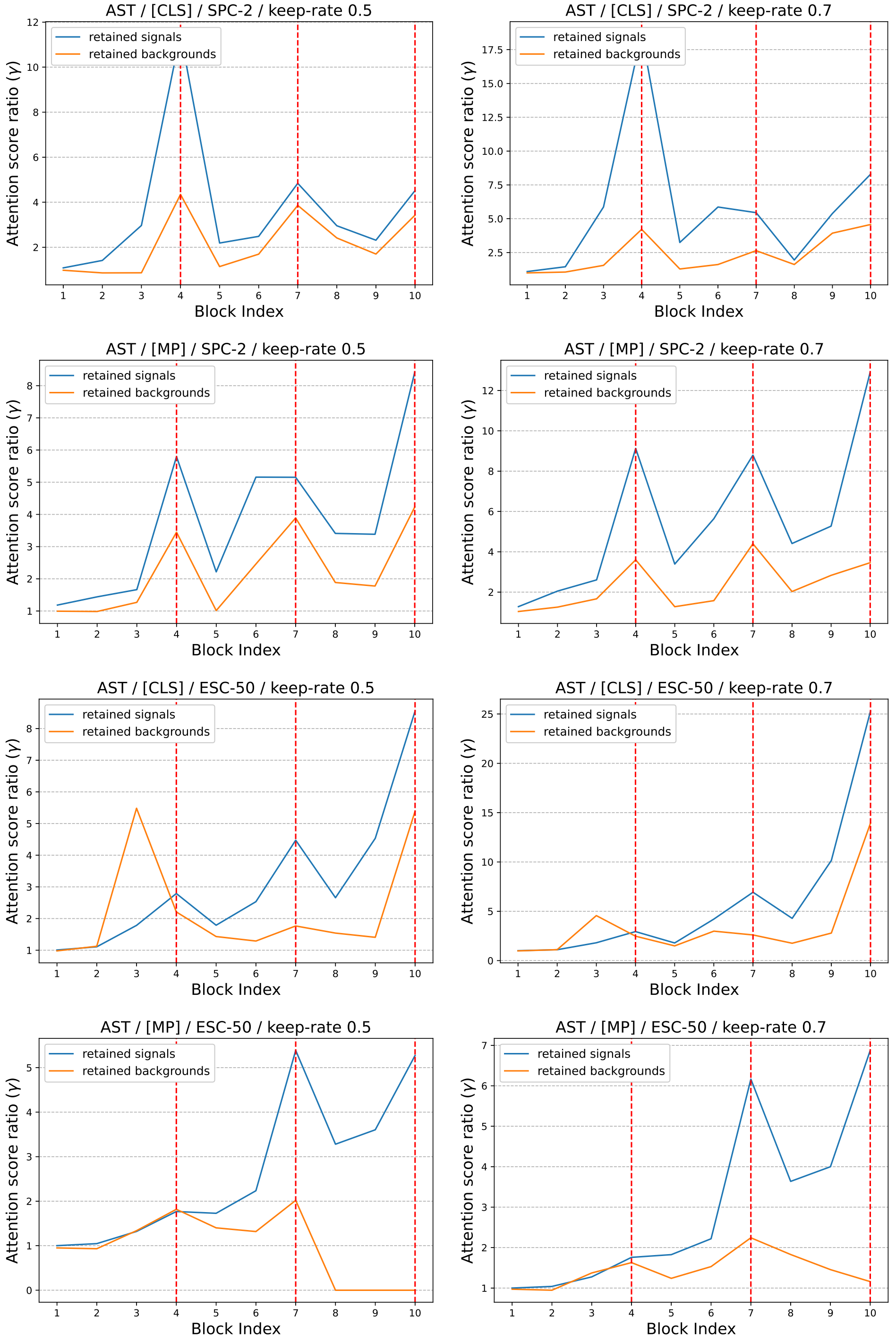


Figure 15: **Ratio of the attention scores of retained and pruned tokens for signals and backgrounds extracted from AST-TopK.** The zero value of retained background tokens indicate that there are no retained tokens from background.

J MORE VISUALIZATION RESULTS OF TOKEN PRUNING USING AUDIOMAE-TOPK

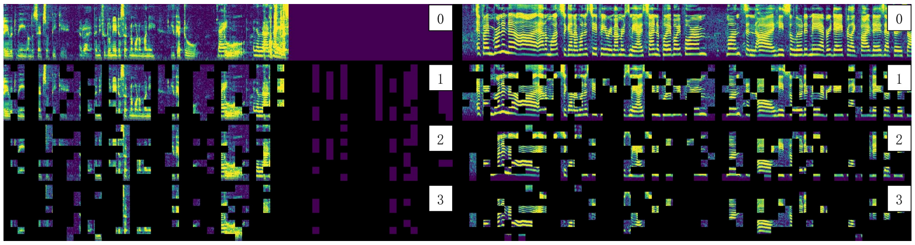


Figure 16: **Visualization of pruned tokens in AudioMAE-TopK, VoxCeleb-1).** *keep-rate* $= 0.5$

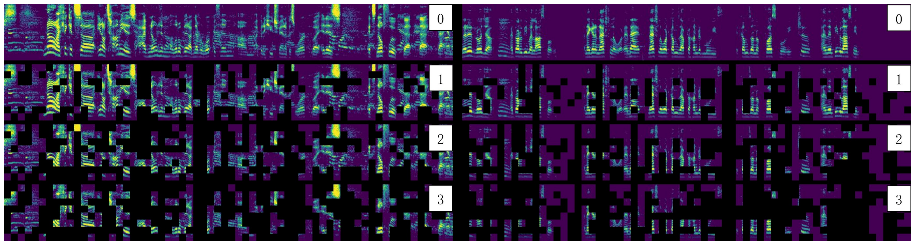


Figure 17: **Visualization of pruned tokens in AudioMAE-TopK, VoxCeleb-1).** *keep-rate* $= 0.7$

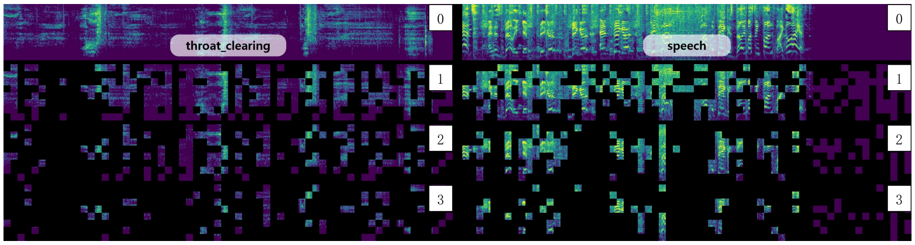


Figure 18: **Visualization of pruned tokens in AudioMAE-TopK, AS-20K).** *keep-rate* $= 0.5$

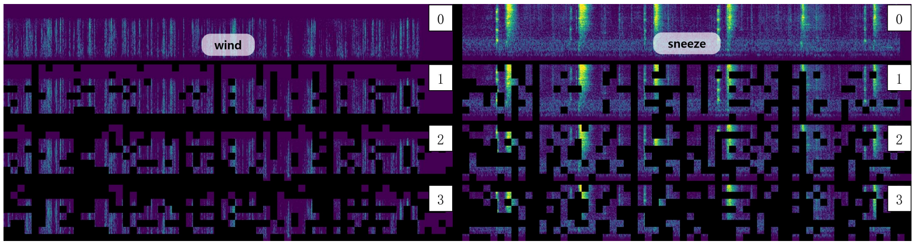


Figure 19: **Visualization of pruned tokens in AudioMAE-TopK, AS-20K).** *keep-rate* $= 0.7$

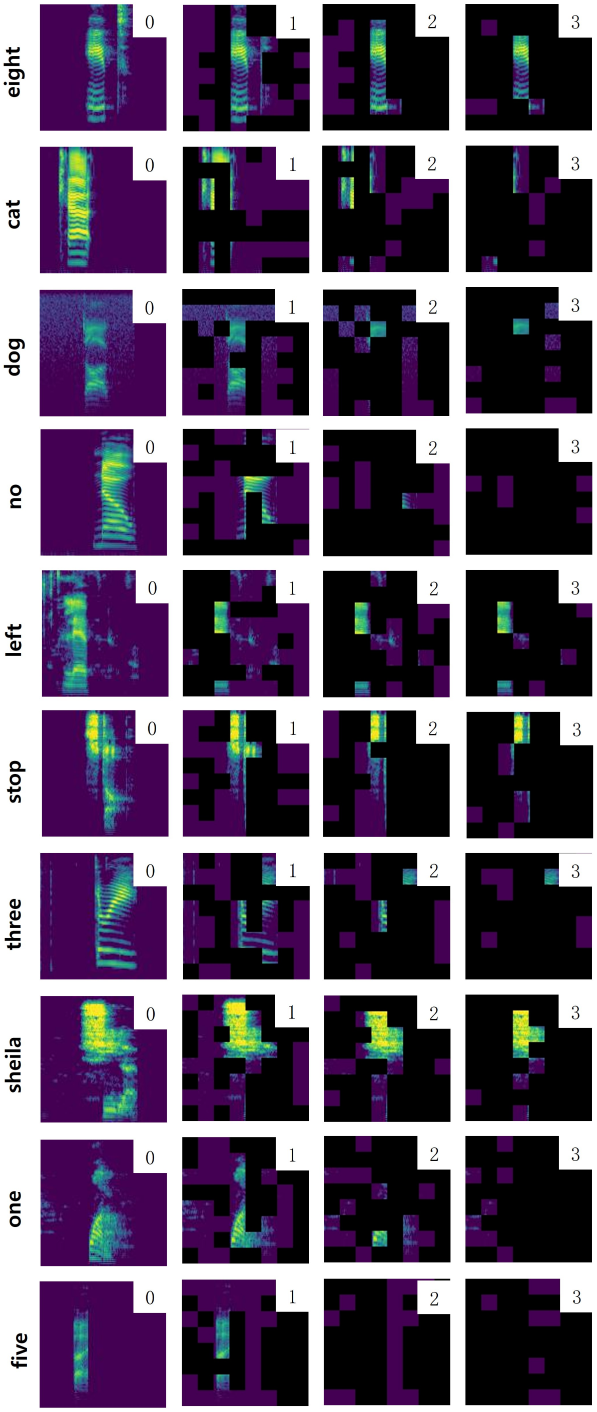
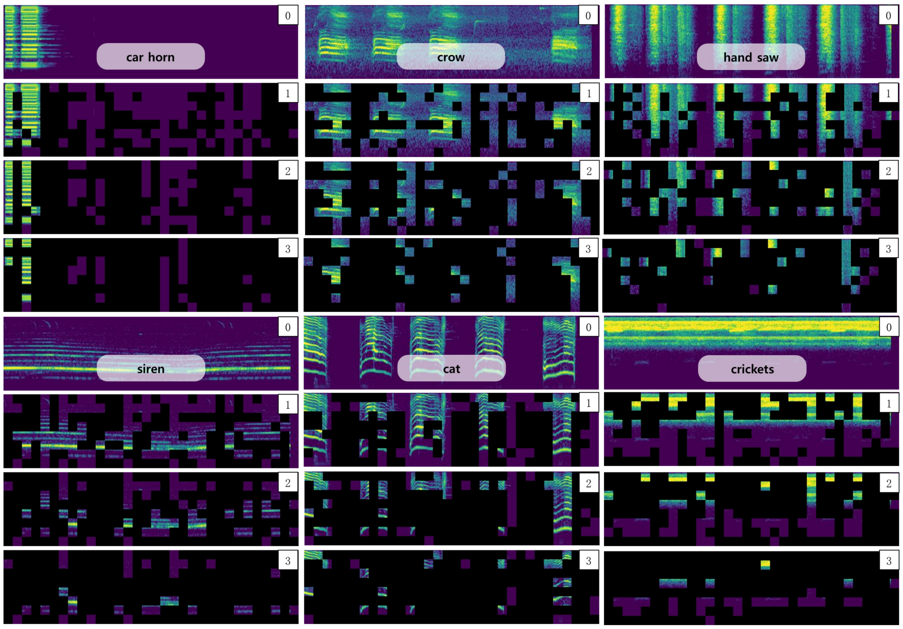
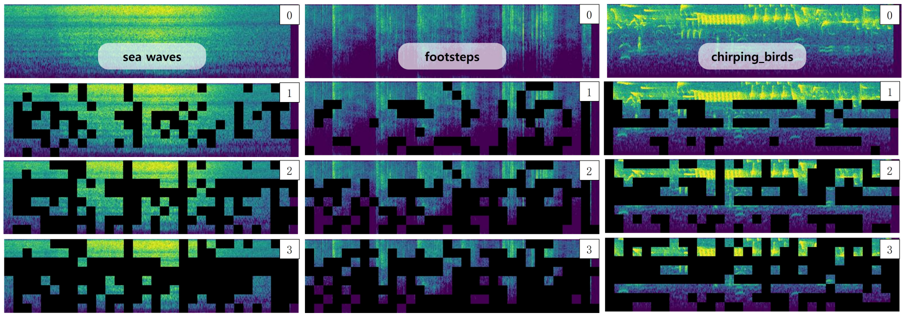


Figure 20: **Visualization of pruned tokens in AudioMAE-TopK, SPC-2).** *keep-rate* $= 0.5$

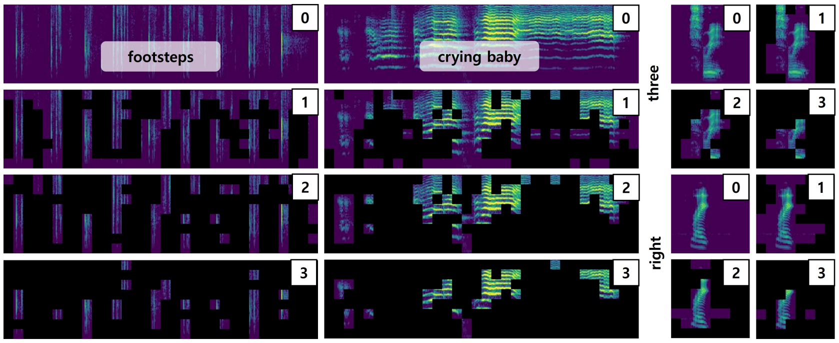


(a) **Visualization of pruned tokens in AudioMAE-TopK, ESC-50).** *keep-rate* $= 0.5$

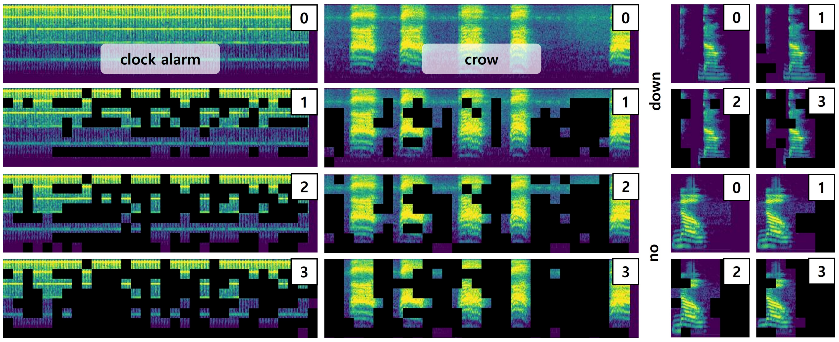


(b) **Visualization of pruned tokens in AudioMAE-TopK, ESC-50).** *keep-rate* $= 0.7$

K MORE VISUALIZATION RESULTS OF TOKEN PRUNING USING AST

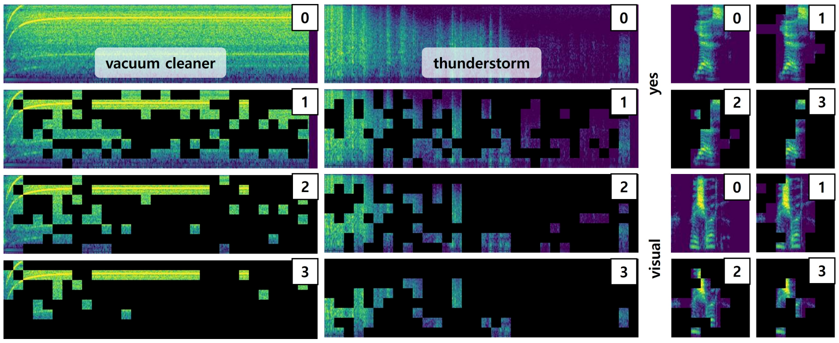


(a) *keep-rate* $= 0.5$

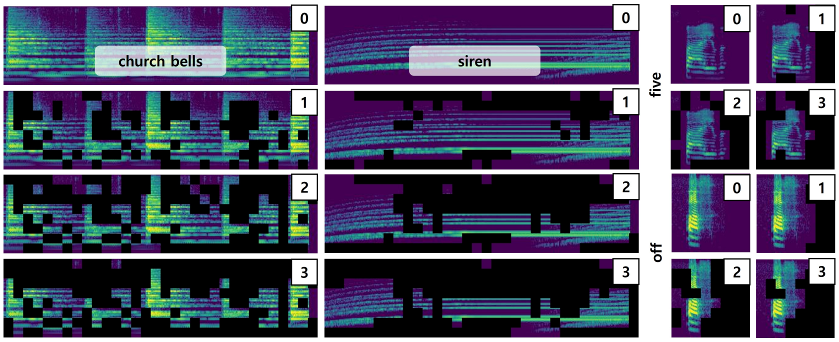


(b) *keep-rate* $= 0.7$

Figure 22: **Visualization of pruned tokens of AST-TopK-CLS with different keep rates.**



(a) *keep-rate* $= 0.5$



(b) *keep-rate* $= 0.7$

Figure 23: **Visualization of pruned tokens of AST-TopK-MP with different keep rates.**