LAST: LOOPED AUDIO SPECTROGRAM TRANSFORMER

Haider Al-Tahan[1] *Sean O'Brien*[2] *Anastasia Razdaibiedina*[3] *N. Apurva Ratan Murty*[1]

[1]Georgia Institute of Technology [2]University of California San Diego
[3]Google DeepMind

ABSTRACT

Increasing depth of transformer models improves recognition, but it comes at a substantial cost. Each additional layer requires more parameters which makes the process computationally inefficient. We ask whether additional processing can focus on integrating features already computed. Looped Audio Spectrogram Transformer (LAST) first processes all tokens, then reuses the same blocks to refine only the class token over fixed audio features, thereby making later passes inexpensive. On AudioSet, ten-pass LAST achieves 0.345 mean average precision, exceeding a twelve-layer sequential transformer by 2.1% relative with 49.4% fewer parameters, 42% fewer multiply-accumulate operations, and 9.8% higher measured throughput. Across separately trained models, increasing the pass count from two to ten improves accuracy while adding only 1.2% computation. Further evaluations show improved robustness to temporal masking and various other auditory augmentations, with better generalization on classification tasks with music, environmental, and events sounds.

***Index Terms*—** audio classification, looped transformers, recursion

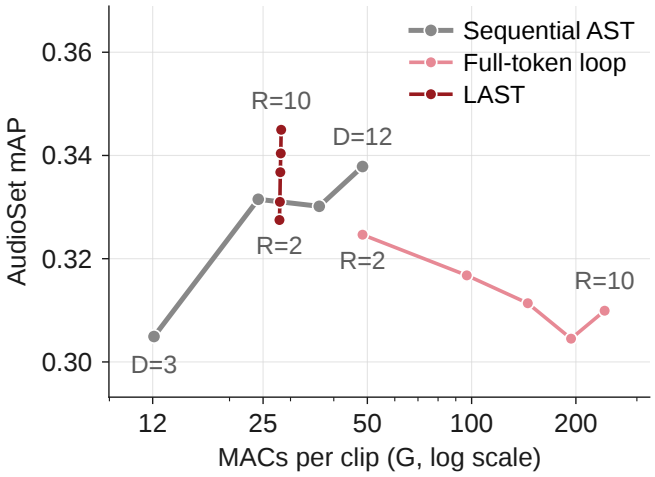

Fig. 1. **LAST outperformed a deeper transformer at lower cost:** AudioSet Computation per clip (x-axis) v/s AudioSet mAP (y-axis) Sequential AST uses $D \in \{3, 6, 9, 12\}$ layers; the recurrent models use $D = 6$ shared blocks and $R \in \{2, 4, 6, 8, 10\}$ passes.

1. INTRODUCTION

Sound is a dynamic signal and recognizing audio requires integrating information across time. Audio Spectrogram Transformers (AST) [1] integrate temporal information over layers by transforming (hundreds of) spectrogram tokens into a class-token representation. Sure enough, increasing model depth improves recognition performance to some degree but the improvement comes at a substantial computational cost. Adding layers means more parameters which makes the process inefficient.

We wondered if there was an alterative way to improve auditory temporal integration. We know that the auditory system in human brains solve the problem of temporal integration via recurrent processing which effectively reuses parameters (neurons). And so we asked if we can we obtain the same depth *effective* benefit of additional layers (depth) without increasing the number of parameters. One possibility is to use looped transformers [2]. Rather than progressing through distinct layers, looped transformers repeatedly apply the same transformer blocks, increasing effective depth (via recurrence) while keeping the number of parameters fixed. Recurrence has shown some benefits for tasks like reasoning in language models [3, 4]. But sound is a more basic challenge. The signal itself unfolds over time and how recurrence contributes to audio recognition is simply not known.

LAST relates to three lines of prior work: (1) recurrence in perceptual models (2) class-token updates over fixed features, and (3) token reduction in audio transformers. Perceiver [5] iteratively cross-attends a latent array to raw image and audio inputs, and LARM [6] loops a speech-recognition encoder that updates every token on each pass. LAST is fundamentally different. It targets the clip-level classification and repeatedly updates *only* the class token while keeping the audio features fixed after a full attention pass. Note that CaiT [7] also refines a class token over fixed features but it uses *separate* attention layers which require new parameters. LAST is more efficient in that it uses the model's existing encoder blocks and therefore does not add new parameters.

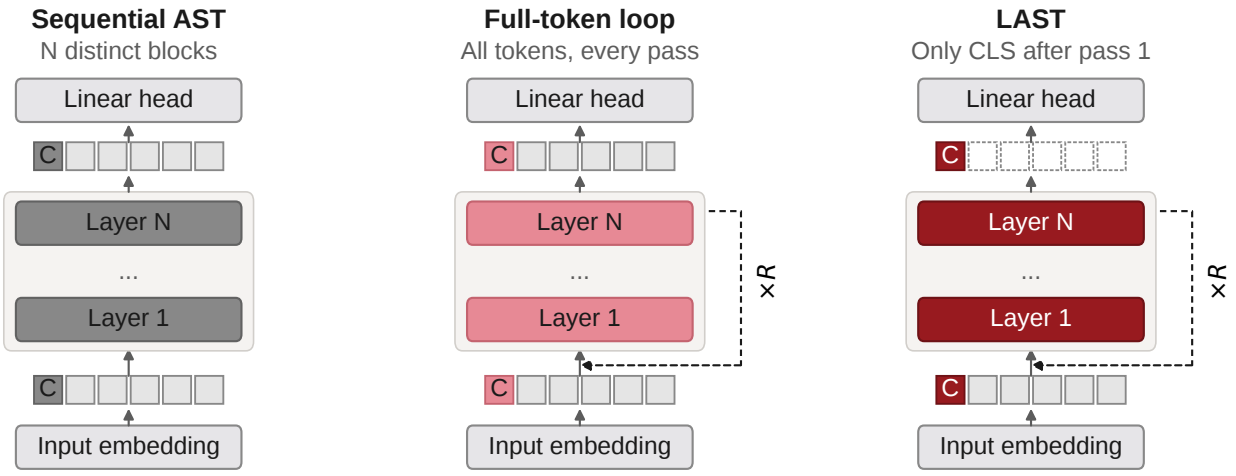

Fig. 2. Sequential AST uses distinct blocks once. Full-token recurrence reuses the same blocks and updates all tokens on every pass. LAST uses one full pass, then updates only the class token using fixed audio features. Recurrent models use six shared blocks.

We seek to adapt this method to the audio domain. Naively, each recurrent pass still updates each spectrogram token and parameter, with computational cost compared to baseline growing linearly in the number of recurrent passes. We observe that, under this naive application, the representations in for the class token tend to be less stable across recurrent steps than those of the other spectrogram tokens. Accordingly, we hypothesize that once informative audio features have been computed, additional processing may be more effectively devoted to integrating information from those pre-computed features rather than repeatedly recomputing the features themselves.

To test this hypothesis, we introduce the Looped Audio Spectrogram Transformer (LAST). LAST first processes the full spectrogram through a set of transformer blocks to construct an audio representation. On subsequent passes, the audio tokens remain fixed while the same transformer blocks repeatedly update only the class token by attending to the fixed audio features. The audio keys and values can therefore be cached and reused across passes, making each additional loop computationally inexpensive. LAST thus increases effective depth while adding neither new parameters nor the cost of repeatedly recomputing the full audio representation.

We found that ten-pass LAST exceeds a twelve-layer sequential baseline by 2.1% relative mAP, with 49.4% fewer parameters and 42% fewer MACs. Increasing the pass count improves LAST at each tested step, whereas additional full-token passes do not improve accuracy. Furthermore, we found LAST models to be more robust and generalizes better on various tasks, retaining higher accuracy on transformations such as temporal masking , and injected noise; and achieves better frozen-feature classification on benchmarks such as ESC-50 [8] and VGGSound [9] than the sequential baseline.

2. METHOD

2.1. Recursion through shared blocks

The encoder takes N log-Mel spectrogram tokens and a learned class (CLS) token of width C, and a linear head predicts clip labels from the final class token. A sequential encoder applies D distinct blocks once. A looped encoder instead reuses the same D blocks for R passes, yielding DR block applications from D sets of weights. Parameters are fixed by D alone, but under full-token recurrence every token is updated on every pass and computation still scales with R.

2.2. Class-token updates in LAST

LAST updates all tokens on the first pass, then fixes the audio tokens. Later passes apply attention and the feed-forward layer (MLP) only to the class token. Each block computes audio keys and values during the second pass and caches them for reuse. Attention and MLP weights are shared. Training remains end to end: gradients reach the first pass through the fixed audio features.

The cost asymmetry follows directly. A full pass costs $O(D(NC^2 + N^2C))$, whereas a cached class-token pass costs $O(D(C^2+NC))$: the quadratic term in N vanishes and the projection term reduces to a single query. Effective depth therefore grows with R at near-constant cost. Figure 2 contrasts the three architectures, and Figure 1 the resulting accuracy–computation trade-off.

3. EXPERIMENTAL SETUP

Training. We train on the 16 kHz archived release of AudioSet [10], with 1,908,644 training clips. All models are trained

from scratch for twelve epochs with one random seed. Inputs have 128 Mel bins and 1024 time frames, divided into 128×2 patches. Models use width $C = 768$, twelve attention heads, an MLP expansion ratio of four, and one class token. We use AdamW with weight decay 0.01, batch size 1024, and learning rate 7.54×10^{-5}. The schedule has 1863 warmup steps and halves the learning rate every three epochs from epoch five. Waveform mixing is applied with probability 0.5, choosing equally between mixup [11] ($\alpha = 10$) and temporal CutMix [12] ($\alpha = 1$). We test sequential depths $D \in \{3, 6, 9, 12\}$ and recurrent models with $D = 6$ shared blocks and $R \in \{2, 4, 6, 8, 10\}$ passes.

3.1. Evaluation

AudioSet. We report mean average precision (mAP) on 18,886 evaluation clips. Results use weights averaged models over epochs 5–12 [13]. We chose this shared window by mean evaluation mAP across runs. Averaging produces one model with no additional inference cost. Robustness and transfer use final-epoch checkpoints.

Throughput. We benchmark inference using bfloat16 and batch size 64.

Transfer benchmarks. We evaluate eight classification tasks: SpeechCommands V2 (SC-V2; keyword spotting) [14], VoxCeleb1 (speaker identification) [15], NSynth (instrument-family recognition) [16], FMA-Small (music genre) [17], ESC-50 [8] and UrbanSound8K (US8K) [18] for environmental sounds, DCASE 2020 Task 1A (acoustic scenes) [19], and VGGSound (audio events) [9]. We report accuracy (%) from one-vs-rest logistic regression with fixed hyperparameters on standardized, frozen class-token features. ESC-50 and UrbanSound8K use their official cross-validation folds. The other datasets use official train/test splits, except NSynth: we train its probe on the official validation split and evaluate on the test split to reduce training cost.

Temporal masking. We zero normalized spectrogram frames in regions selected by descending class-token attention or uniformly at random. Attention scores are computed separately for each model from the unmasked input's final block, averaged over heads.

Waveform corruption. We test 38 corruption settings across twelve families (Figure 4). These change noise, timing, frequency, reverberation, or amplitude before spectrogram extraction. Each setting uses the same 4,096 clips. We compare sequential $D = 12$ with LAST at $D = 6$, $R = 10$, using final-epoch checkpoints.

4. RESULTS

4.1. Accuracy and computation

We first ask whether effective depth obtained by *reusing* blocks can substitute for depth obtained by *adding* them, and at what cost. Three model families are trained under an identical recipe: sequential AST at $D \in \{3, 6, 9, 12\}$, and two recurrent variants with $D = 6$ shared blocks and $R \in \{2, 4, 6, 8, 10\}$ passes. Two findings emerge. First, LAST improves at every pass count we test while its cost stays nearly flat. Second, this gain is specific to restricting later passes to the class token: full-token recurrence, given the same blocks and the same pass counts, gets *worse* with depth while costing far more.

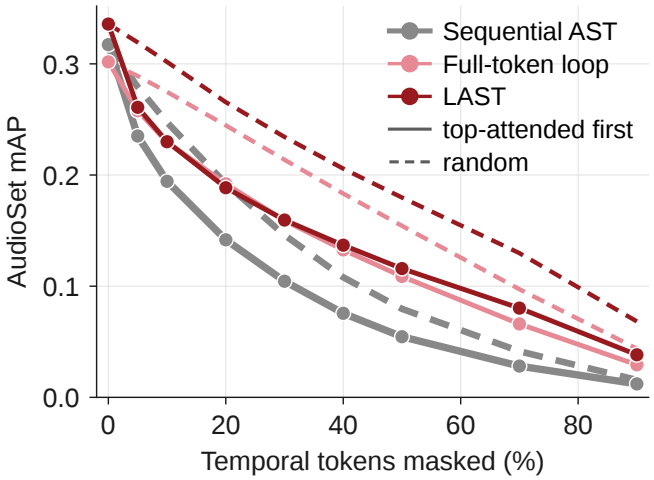

Fig. 3. **LAST degraded more gracefully under temporal masking:** AudioSet mAP under temporal masking. Masking zeros the corresponding normalized spectrogram frames. Models: sequential $D = 12$ and recurrent $D = 6$, $R = 10$.

Adding blocks works, but expensively. Increasing sequential depth from $D = 3$ to $D = 12$ raises mAP from 0.3049 to 0.3379, at the cost of growing parameters from 22.3M to 86.1M and computation from 12.1 to 48.4 GMACs. Against the parameter-matched sequential $D = 6$ baseline, LAST wins once $R \geq 6$, and its mAP rises at every tested pass count, from 0.3275 at $R = 2$ to 0.3450 at $R = 10$. Because later passes update only the class token and reuse cached audio keys and values, these eight extra passes add just 1.2% computation. The comparison against the deeper baseline is therefore favourable on all three axes at once: at $R = 10$, LAST reaches 0.3450 mAP against 0.3379 for sequential $D = 12$ — a 2.1% relative improvement — with 49.4% fewer parameters, 42% fewer MACs, and 9.8% higher measured throughput.

The benefit depends on how recurrence is applied. With the same six shared blocks, full-token recurrence falls from 0.3247 mAP at $R = 2$ to 0.3099 at $R = 10$, despite a fivefold increase in computation. LAST's advantage at matched R grows from 0.0028 mAP at $R = 2$ to 0.0350 at $R = 10$, where full-token recurrence requires approximately 8 times as many MACs. These results favor class-token recurrence over full-token recurrence under the tested training recipe. Each data point in Figure 1 is a model that is trained separately, so these results do not establish whether an already trained model benefits from additional inference passes.

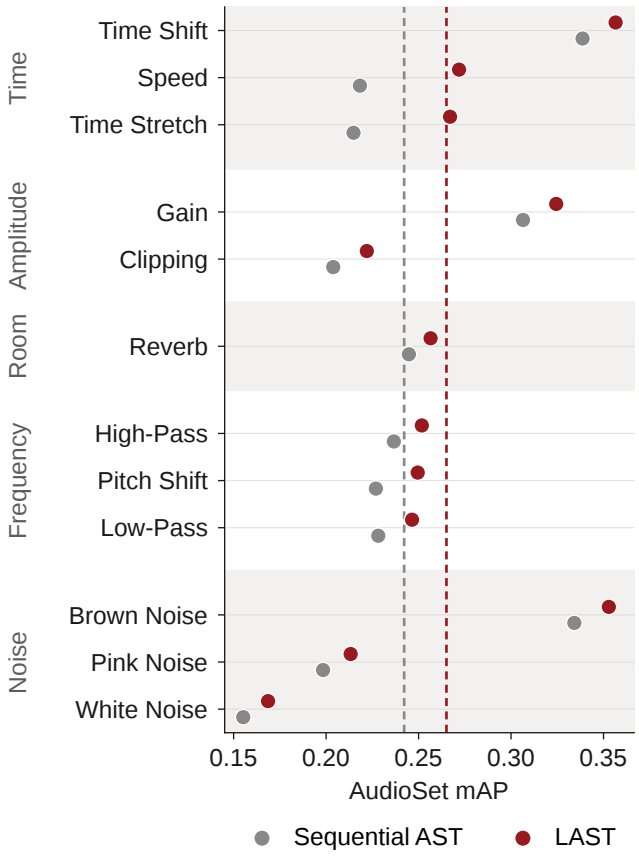

Fig. 4. **LAST improved performance across the board:** AudioSet mAP under waveform corruption. Each dot is the mean over settings within one family; each dashed line is the mean across the twelve families. Models: sequential $D = 12$ and LAST $D = 6$, $R = 10$.

4.2. Robustness

Clean AudioSet mAP measures only the matched condition. If repeated class-token passes genuinely integrate evidence distributed across the clip, rather than sharpening the model's reliance on a few salient frames, the representation should degrade gracefully when that evidence is removed or distorted. We probe this from two directions: temporal masking *removes* information, and waveform corruption *alters* it while leaving the input complete. LAST is more robust in both, its margin over sequential AST exceeds the two models' gap on clean audio, and on corruptions it leads in every family tested.

Temporal masking. We progressively zero normalized spectrogram frames, selecting regions either by descending class-token attention or uniformly at random. LAST retains more accuracy than sequential AST under both attention-based and random temporal masking (Figure 3). It achieves the highest mean mAP among the three models. Integrating each curve over 0–90% masking and dividing by that range gives mean mAP of 0.1383 versus 0.0905 for LAST and sequential AST under attention-based masking, and 0.1970 versus 0.1187 under random masking.

Table 1. **LAST's features transferred better on most tasks, with half the parameters:** Frozen-feature classification accuracy (%) using final-epoch encoders and the protocols in Section 3.1.

	Sequential AST	Looped, full-token	**LAST**
Depth	12	6	6
Recurrences	–	2	10
Parameters (M)	86.1	43.5	43.5
MACs (G)	48.4	48.4	**28.2**
Throughput (clips/s)	571	572	**628**
AudioSet mAP	0.3379	0.3247	**0.3450**
Voice & Speech			
SC-V2	**71.9**	71.6	69.8
VoxCeleb	3.6	**4.3**	4.0
Music			
NSynth	**96.9**	96.1	96.8
FMA	43.5	42.1	**50.9**
Environmental & Scenes			
ESC-50	84.1	83.6	**85.7**
US8K	**78.1**	76.8	77.8
DCASE	46.7	**48.6**	48.3
VGGSound	34.9	34.6	**36.3**

Waveform corruption. Masking withholds information outright; waveform corruption instead perturbs the signal before spectrogram extraction, so the model receives a complete but distorted input. Across 38 settings spanning twelve families — noise, timing, frequency, reverberation, and amplitude — LAST attains higher mAP in every family (Figure 4). Averaging within each family and then across families gives 0.2651 for LAST against 0.2422 for sequential $D = 12$, a gap of 0.0229. The largest gaps occur under speed perturbation and time stretching, compared with 0.012 to 0.023 across the remaining families.

4.3. Frozen-feature transfer

Transfer-set benchmarks asks a different question: is the class token, having been shaped by repeated integration, a better general-purpose clip descriptor? We freeze the class-token features and fit one-vs-rest logistic regression with fixed hyperparameters on eight tasks spanning speech, music, environmental sound, and acoustic scenes (Table 1). LAST outperforms sequential $D = 12$ on five of eight transfer benchmarks. The largest gain is on FMA-Small, from 43.5% to 50.9%; LAST also improves ESC-50 and VGGSound accuracy. DCASE accuracy rises from 46.7% to 48.3%.

5. REFERENCES

[1] Yuan Gong, Yu-An Chung, and James Glass, "AST: Audio spectrogram transformer," in *Proc. Interspeech*, 2021, pp. 571–575.

[2] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser, "Universal transformers," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2019.

[3] Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein, "Scaling up test-time compute with latent reasoning: A recurrent depth approach," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2025, vol. 38, pp. 41340–41391.

[4] Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J. Reddi, "Reasoning with latent thoughts: On the power of looped transformers," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2025.

[5] Andrew Jaegle, Felix Gimeno, Andy Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira, "Perceiver: General perception with iterative attention," in *Proc. Int. Conf. Machine Learning (ICML)*, 2021.

[6] Yacouba Kaloga, Shashi Kumar, Shakeel A. Sheikh, Driss Khalil, Petr Motlicek, and Ina Kodrasi, "Test-time compute scaling for ASR with depth-conditioned looped transformers," *arXiv preprint arXiv:2606.04678*, 2026.

[7] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou, "Going deeper with image transformers," in *Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV)*, 2021.

[8] Karol J. Piczak, "ESC: Dataset for environmental sound classification," in *Proc. ACM Int. Conf. Multimedia*, 2015, pp. 1015–1018.

[9] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman, "VGGSound: A large-scale audio-visual dataset," in *Proc. IEEE ICASSP*, 2020, pp. 721–725.

[10] Jort F. Gemmeke, Daniel P. W. 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 *Proc. IEEE ICASSP*, 2017, pp. 776–780.

[11] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2018.

[12] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV)*, 2019.

[13] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson, "Averaging weights leads to wider optima and better generalization," in *Proc. Conf. Uncertainty in Artificial Intelligence (UAI)*, 2018.

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

[15] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in *Proc. Interspeech*, 2017, pp. 2616–2620.

[16] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan, "Neural audio synthesis of musical notes with WaveNet autoencoders," in *Proc. Int. Conf. Machine Learning (ICML)*, 2017.

[17] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson, "FMA: A dataset for music analysis," in *Proc. Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2017.

[18] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello, "A dataset and taxonomy for urban sound research," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 1041–1044.

[19] Toni Heittola, Annamaria Mesaros, and Tuomas Virtanen, "Acoustic scene classification in DCASE 2020 challenge: Generalization across devices and low complexity solutions," in *Proc. Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2020, pp. 56–60.