Supplementary Material for Masked Autoencoders that Listen

Po-Yao Huang[1] **Hu Xu**[1] **Juncheng Li**[2] **Alexei Baevski**[1]
Michael Auli[1] **Wojciech Galuba**[1] **Florian Metze**[1] **Christoph Feichtenhofer**[1]

[1]Meta AI [2]Carnegie Mellon University

Outline

The appendix is organized as follows: In §A, we first demonstrate additional audible visualizations with anonymous URL links. In §B, we provide the complete experimental details and hyperparameter configurations for pre-training and fine-tuning on each dataset. Then in §C, we conduct extra experiments on ESC-50 (§C.1) with additional supervised pre-training on AudioSet to complete the comparison with the models marked with † in Table 2 of the main paper. We then study a case how Audio-MAE could be applied to a practical speech generation task (§C.2); and share some negative results and insights on directions we tried that did not work well (§C.3). Finally, we discuss the limitations (§D) of Audio-MAE.

A Additional Reconstruction Details and Results by Audio-MAE Decoder

Fig. 1 illustrates additional reconstruction results on the AudioSet-2M *eval* set. Audible examples are under the anonymous links, accessible by clicking on respective 1 2 3. (1 is the ground truth reference, 2 is the masked input for Audio-MAE, and 3 is the reconstruction output by Audio-MAE.)

We use an Audio-MAE model with a ViT-L encoder and a 16-layer decoder with local attention for visualization. The model is trained under 80% unstructured (random) masking on AudioSet. We inverse Mel-spectrograms and exploit the Griffin-Lim [1] algorithm to reconstruct waveform. There could be perceivable artifacts due to imperfect phase estimation in [1]. Note that the default masking ratio in Fig. 1 is 70% for better visualization. We also show reconstruction results under 80% masking ratio in Fig. 1e-1h for comparison.

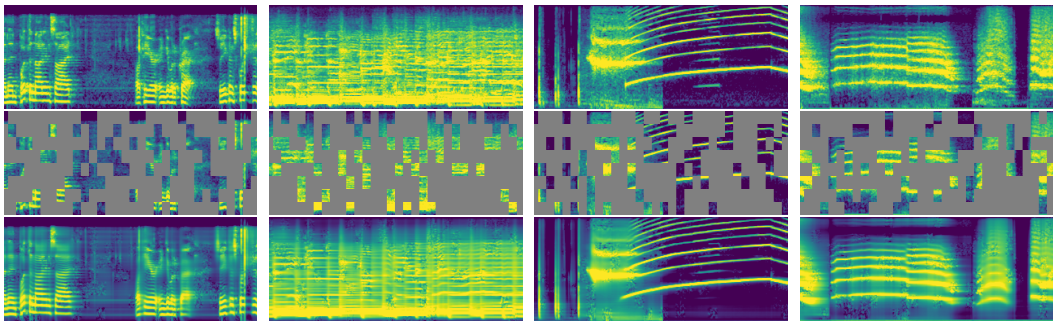
Comparing 2 and 3 under the each caption in Fig. 1, even with 70%-80% masking ratio, Audio-MAE can still create reasonable reconstructions. Music and event sound are easier for Audio-MAE due to their relatively predictable spectrogram patterns. For example, the repeating tempos across time domain (*e.g.*, the music in Fig. 1b and Fig. 1l) and the harmonics across frequency domain (*e.g.*, the siren in Fig. 1c and the trumpeting elephant in Fig. 1d) are very well reconstructed. Speech recordings are more challenging as shown in Fig. 1a and Fig. 1e.

In most cases, Audio-MAE successfully restores audio from masked/corrupted inputs. With these encouraging results, we envision that Audio-MAE can also be applied to other speech generation tasks and qualitatively case-study an application in §C.2.

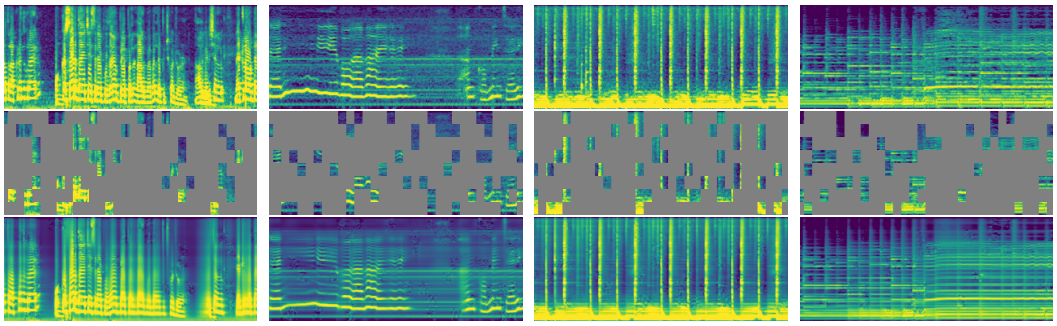
B Experimental Details and Hyperparameter Settings

In this section we provide additional experimental details. For audio recordings in each dataset, we pre-process all of them into mono channel under 16K sampling rate for simplicity and consistency between pre-training and fine-tuning tasks. Note that their native sampling rate may not be 16K (there are many 8K or higher sampling rate recordings in AudioSet. Also, video compression by YouTube may up-samples or down-samples the audio tracks of user-uploaded videos). During data loading, we

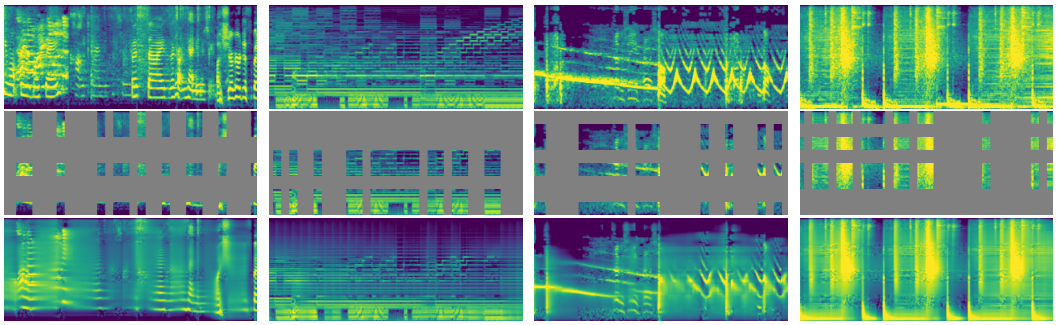
36th Conference on Neural Information Processing Systems (NeurIPS 2022).



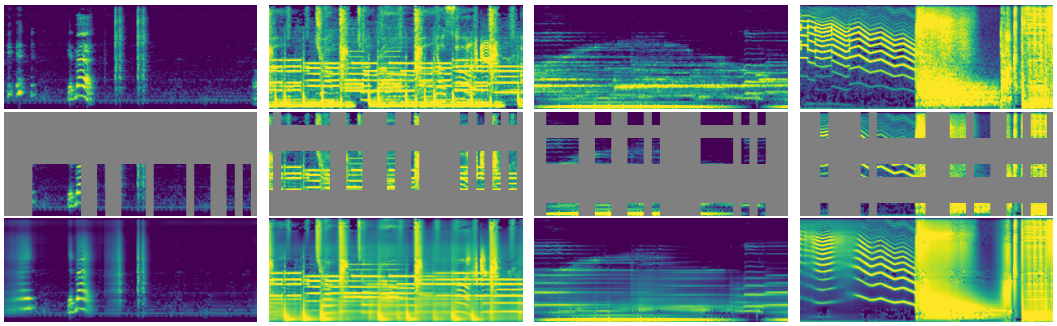
(a) 70% Unstructured 1 2 3 (b) 70% Unstructured 1 2 3 (c) 70% Unstructured 1 2 3 (d) 70% Unstructured 1 2 3



(e) 80% Unstructured 1 2 3 (f) 80% Unstructured 1 2 3 (g) 80% Unstructured 1 2 3 (h) 80% Unstructured 1 2 3



(i) 70% Structured 1 2 3 (j) 70% Structured 1 2 3 (k) 70% Structured 1 2 3 (l) 70% Structured 1 2 3



(m) 70% Structured 1 2 3 (n) 70% Structured 1 2 3 (o) 70% Structured 1 2 3 (p) 70% Structured 1 2 3

Figure 1: **Additional spectrogram reconstruction visualizations on the AudioSet *eval* set**. Column-wise type: speech, music, event, others. Masking type: (a-h) unstructured (random); (i-p) structured (time+frequency). Masking ratio: 80% for (e-h) and the rest are 70% . In each group, we show the original spectrogram (1, top), masked input (2, middle), and Audio-MAE output (3, bottom). The spectrogram size is 1024×128; patch size is 16×16. Each sample has 64×8=512 patches with either 154 (for 70% masked) or 102 (for 80% masked) patches being visible to Audio-MAE. Please click on corresponding (1 2 3) for audible *.wav*s.

Configuration	pre-training AS-2M PT	fine-tuning AS-2M	AS-20K	ESC [2]	SPC-2 [3]	SPC-1	SID [4]
Optimizer	AdamW [5]						
Optimizer momentum	$\beta_1 = 0.9, \beta_2 = 0.95$						
Weight decay	0.0001						
Base learning rate	0.0002	0.0002†	0.001	0.001	0.001	0.001	0.001
Learning rate schedule	half-cycle cosine decay [6]						
Minimum learning rate	0.000001						
Gradient clipping	None						
Warm-up epochs	3	20	4	4	4	1	4
Epochs	32	100	60	60	60	10	60
Batch size	512	512	32	64	256	256	64
GPUs	64	64	4	4	4	4	4
Weighted sampling	False	True	False	False	False	False*	False
Weighted sampling size	-	200,000	-	-	-	-	-
Augmentation	R	R	R	R	R+N	R+N	R+N
SpecAug [7] (time/frequency)	-	192/48	192/48	96/24	48/48	48/48	192/48
Drop path [8]	0.0	0.1	0.1	0.1	0.1	0.1	0.1
Dropout [9]	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Mixup [10]	0.0	0.5	0.5	0.0	0.5	0.5	0.0
Multilabel	n/a	True	True	False	False	False	False
Loss Function	MSE	BCE	BCE	CE	BCE	BCE	CE
Dataset Mean for Normalization	-4.268	-4.268	-4.268	-6.627	-6.846	-6.702	-6.370
Dataset Std for Normalization	4.569	4.569	4.569	5.359	5.565	5.448	3.074

Table 1: **Pre-training (PT) and Fine-tuning (FT) hyperparameters**. For augmentation, R: sampling random starting points with cyclic rolling in time; N: adding random noise (signal-to-noise ratio (SNR): 20dB) to spectrograms. For loss functions, BCE: binary cross entropy loss (for multi-label datasets or when using mixup [10]); CE: cross-entropy loss, MSE: mean square error loss. *: We repeat and balance each class to 50% of the size of the unknown class. †: For ViT-S, We use a learning rate of 0.0005 on AS-2M FT and 0.002 on AS-20K FT as we find larger learning rates work better for ViT-S encoder.

pad or trim the audio length (in seconds) on each dataset as follows: AudioSet: 10, ESC: 5, SPC-1 and SPC-2: 1, SID: 10 seconds. We use a window of 25 ms with a hop length of 10 ms to transform waveform into 128 mel-bank features. The resulting input shapes are: AudioSet: $1 \times 1024 \times 128$, ESC: $1 \times 512 \times 128$, SPC: $1 \times 128 \times 128$, SID: $1 \times 1024 \times 128$. With different input shapes and audio types, we adjust the hyperparameters and data augmentation for each task respectively. We summarize the pre-training (AS-2M PT) and fine-tuning details on each dataset in Table 1.

We adopt most of the default hyper-parameters used in MAE [11]. Note that the effective learning rate (lr_{eff}) depends on the base learning rate (lr_{base}) and the batch size. Precisely, $lr_{\text{eff}} = lr_{\text{base}} * \frac{\text{batch size}}{256}$. When the dataset is multi-label or the mixup [10] augmentation is enabled, we use binary cross-entropy loss (BCE) as the fine-tuning objective without label smoothing [12].We also experimented using strong data augmentations (*e.g.*, mixup [10], SpecAug [10], and CutMix [13]) for pre-training but found the resulting performance similar or worse (especially for CutMix which resulted in ~0.5 mAP degrade in AudioSet-2M). Therefore we discard these strong data augmentations in the pre-training phase by default.

To perform importance sampling when fine-tuning on the unbalanced AudioSet-2M, following prior works, we apply a weighted sampler. We set the probability of sampling a sample proportional to the inverse frequency of its labels, where the label frequency is estimated over the training set. Specifically, for a instance i in a dataset $\mathbf{D}$ with a label pool $\mathbf{C}$, its sampling weight is proportional to $\sum_{c_i \in \mathbf{C}} w_c$, where $w_c = \frac{1000}{\sum_{i \in \mathbf{D}} c_i + \epsilon}$ and $\epsilon = 0.01$ is set to avoid underflow in majority classes as in [14]. In each fine-tuning epoch on AS-2M, we sample 200K instances (~10% of AudioSet-2M) without replacement in avoidance of duplicated samples in a batch and repeating samples within an epoch. We fine-tune for 100 epochs, which aggregate to ~10 full epochs of AudioSet-2M. Proper normalization for audio is important to avoid pre-training fine-tuning discrepancy. We use the training split of each end task to estimate dataset-wise mean and standard deviation The code, scripts, and pre-trained models for reproducibility are at https://github.com/facebookresearch/AudioMAE.

C Additional Experiments

In this section, we extend our experimental investigation of Audio-MAE to include additional results that are not covered in the main paper. First (§C.1), on ESC-50, we report and compare model performance under an additional round of supervised pre-training on labeled AudioSet-2M (models marked with † in Table 2 of the main paper). Second (§C.2), we include additional qualitative results on packet loss concealment (PLC) as a preliminary case study on practically useful downstream tasks for the *decoder* in Audio-MAE, and demonstrate its potential impact for generative applications. Third (§C.3), we share some negative results when we tried incorporating contrastive objectives for Audio-MAE. Our findings suggest that using reconstruction objective alone is sufficient.

C.1 ESC-50 with AudioSet-2M Supervised Pre-training

ESC-50 is designed for environmental sound classification. Besides the pre-training setup introduced in the original paper, we further study a widely compared setup where the models are additionally supervisedly pre-trained with AudioSet data and labels before fine-tuning on ESC-50. Table 2 summarizes the results under this setup where our Audio-MAE achieves state of the art accuracy with the additional AudioSet-2M supervised pre-training. Note that our model is still audio-only and uses *no* ImageNet data (IN-SL).

Model	Backbone	Pre-training	ESC-50 FT
ERANN [15]	CNN	AS-SL	96.1
PANN [16]	CNN	AS-SL	94.7
AST [14]	DeiT-B	IN-SL, AS-SL	95.6
HTS-AT [17]	Swin-B	IN-SL, AS-SL	97.0
PASST [18]	DeiT-B	IN-SL, AS-SL	96.8
Audio-MAE (global)	ViT-B	AS-SSL, AS-SL	96.9
Audio-MAE (local)	ViT-B	AS-SSL, AS-SL	**97.4**

Table 2: **Comparison with other state-of-the-art models on ESC-50** with an additional round of supervised pre-training on AudioSet (AS-SL). SSL: self-supervised learning. We gray-out the models with out-of-domain pre-training on ImageNet (IN).

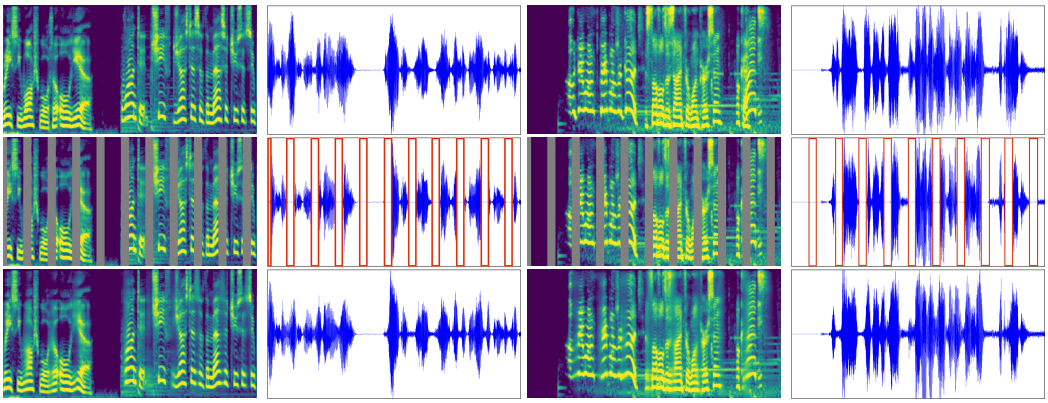
C.2 Qualitative Results for a practical generation task

Packet Loss Concealment (PLC) is a widely deployed technique to alleviate side effects from missing or corrupted packets in Voice over IP (VoIP) applications (*e.g.*, video conferencing, Bluetooth earbuds, wireless virtual reality headset, *etc.*) When an encoded speech is sent as a sequence of VoIP packets over a network, these packets may get lost or be corrupted during the transmission, resulting in undesirable low quality speech. To this end, various PLC techniques has been developed. The recent approaches substitute the corrupted waveform segments by either replacing the corrupted waveform segments with other intact segments base on the acoustic pitch detected, or via inpainting with RNN-based [19], CNN-based [20], or autoencoding-based [21, 22] reconstruction.

In this section, we qualitatively demonstrate how Audio-MAE could potentially be applied for PLC to recover corrupted waveform segments with its encoder-decoder architecture. In Fig. 2, we simulate two time-corrupted speech recordings by masking speech in time and perform reconstruction with Audio-MAE. In practice, a PLC system may exploit packet checksums to identify corrupted or missing packets and mask them. The PLC problem then can be viewed as a special case (time-only, structured masking) of Audio-MAE. As shown in both cases, the Audio-MAE decoder produces reasonable speech reconstruction. We leave the in-depth study and analysis of generative tasks (*e.g.* PLC and speech bandwidth expansion (BWE) [23, 4]) as the future work.

C.3 Negative Results: Directions that did not work well

Additional Contrastive Objective We examined using additional contrastive objectives in the pre-training phase but do not find them helpful empirically. Similar to SS-AST [24] and Wave2vec



(a) Speech one (Freq./Time) 1 2 3 (b) Speech two (Freq./Time) 1 2 3

Figure 2: **Qualitative Results for Packet Loss Concealment with Audio-MAE Decoder**. Simulations of 25% packet loss rate in time for two speech recordings. In each group, we show the original spectrogram(left) and time(right) sequence (1, top), corrupted input with packet loss (2, middle), and Audio-MAE restoration (3, bottom). The spectrogram size is 1024×128; patch size is 16×16. Please click (1 2 3) for audible *.wav*s.

2.0 [25], we apply InfoNCE [26] loss over masked tokens of an instance. Specifically, let $\mathbf{x}_i, i = 1 \dots N$ denotes the values of i-th masked spectrogram patch where N is the number of masked patches in an instance. (*e.g.*, rounded $N = 102$ under 80% masking over 64×8 spectrogram patches of a 10-second audio recording.) And let $\mathbf{c}_i$ denotes its corresponding contextualized embedding projected by a separated decoder head. We investigate the following contrastive objective:

$$L_c = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{\mathbf{c}_i^T\mathbf{x}_i}}{\sum_{j=1}^{N}e^{\mathbf{c}_i^T\mathbf{x}_j}}. \tag{1}$$

Intuitively, L_c draws closer patches with their contextualized embeddings (positive pairs) at each masked position while contrasting and pushing away mismatched ones (negative pairs) from all masked patches. For the reconstructive objective, let $\hat{\mathbf{x}}_i, i = 1 \dots N$ be the reconstruction of i-th masked spectrogram patch generated by the reconstruction head of our Audio-MAE decoder. The original reconstruction objective L_r in Audio-MAE is formally defined as:

$$L_r = \frac{1}{N}\sum_{i=1}^{N}(\hat{\mathbf{x}}_i - \mathbf{x}_i)^2. \tag{2}$$

We consider three setups: (1) Using the reconstructive objective (L_r) alone (the default setup); (2) using the contrastive objective (L_c) alone; (3) multi-tasking with both the reconstructive and contrastive objectives ($L_r + \alpha L_c$), where α is the hyper-parameter that balances two objectives.

Table 3 shows the results: We see that the reconstruction objective L_r alone is sufficient and yields the best performance. Empirically, we do not observe improvement with contrastive objectives alone or under the multi-task setup (the best α is 0.2 in our experiments). L_c and L_r do not work complementarily in Audio-MAE.

Objective	AS-20K	AS-2M
Reconstruction (L_r)	**37.1**	**47.3**
Contrastive (L_c)	36.4	46.6
Contrastive + Reconstruction ($L_r + \alpha L_c$)	36.8	46.8

Table 3: **Impact of contrastive objective**.

D Limitations

We think there are few direct limitations of this work. The data scale is one of them. AudioSet used by Audio-MAE is around two orders of magnitude smaller than the text corpus used in the language [27, 28, 29] counterparts. Another limitation is duration of each sample: the 10-second recordings in AudioSet are short and thus distant temporal dependencies in audio may not be properly learned yet. Further, as AudioSet is unbalanced and there are many audio types beyond the 527 classes annotated in AudioSet, Audio-MAE could be sub-optimal when transferring to tasks concerning rare or unseen audio events. Lastly, while Audio-MAE has greatly improved the efficiency of large-scale self-supervised learning, modeling lengthy audio and high-dimensional data with Transformers is computationally demanding.

References

[1] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[2] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*. ACM Press, 2015, pp. 1015–1018.

[3] P. Warden, "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition," *ArXiv e-prints*, Apr. 2018.

[4] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Comput. Speech Lang.*, vol. 60, 2020.

[5] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[6] ——, "SGDR: stochastic gradient descent with warm restarts," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[7] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *ArXiv*, vol. abs/1904.08779, 2019.

[8] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, ser. Lecture Notes in Computer Science, vol. 9908. Springer, 2016, pp. 646–661.

[9] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[10] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[11] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," *arXiv preprint arXiv:2111.06377*, 2021.

[12] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019, pp. 4696–4705.

[13] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019, pp. 6022–6031.

[14] Y. Gong, Y. Chung, and J. R. Glass, "AST: audio spectrogram transformer," in *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*. ISCA, 2021, pp. 571–575.

[15] S. Verbitskiy, V. Berikov, and V. Vyshegorodtsev, "Eranns: Efficient residual audio neural networks for audio pattern recognition," *arXiv preprint arXiv:2106.01621*, 2021.

[16] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 2880–2894, 2020.

[17] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection," *arXiv preprint arXiv:2202.00874*, 2022.

[18] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," *CoRR*, vol. abs/2110.05069, 2021.

[19] B. Lee and J. Chang, "Packet loss concealment based on deep neural networks for digital speech transmission," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 24, no. 2, pp. 378–387, 2016.

[20] J. Lin, Y. Wang, K. Kalgaonkar, G. Keren, D. Zhang, and C. Fuegen, "A time-domain convolutional recurrent network for packet loss concealment," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 7148–7152.

[21] Y. Chang, K. Lee, P. Wu, H. Lee, and W. H. Hsu, "Deep long audio inpainting," *CoRR*, vol. abs/1911.06476, 2019.

[22] A. Marafioti, N. Perraudin, N. Holighaus, and P. Majdak, "A context encoder for audio inpainting," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 27, no. 12, pp. 2362–2372, 2019.

[23] B. Liu, J. Tao, Z. Wen, Y. Li, and D. Bukhari, "A novel method of artificial bandwidth extension using deep architecture," in *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*. ISCA, 2015, pp. 2598–2602.

[24] Y. Gong, C.-I. Lai, Y.-A. Chung, and J. R. Glass, "Ssast: Self-supervised audio spectrogram transformer," *ArXiv*, vol. abs/2110.09784, 2021.

[25] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[26] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *CoRR*, vol. abs/1807.03748, 2018.

[27] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019, pp. 4171–4186.

[28] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019.

[29] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.