THINKING WHILE LISTENING: SIMPLE TEST TIME SCALING FOR AUDIO CLASSIFICATION

Prateek Verma *Mert Pilanci*

Department of Electrical Engineering
Stanford University
Stanford, CA 94305, USA

ABSTRACT

We propose a framework that enables neural models to "think while listening" to everyday sounds, thereby enhancing audio classification performance. Motivated by recent advances in the reasoning capabilities of large language models, we address two central questions: (i) how can thinking be incorporated into existing audio classification pipelines to enable reasoning in the category space and improve performance, and (ii) can a new architecture be designed from the ground up to support both thinking and test-time scaling? We demonstrate that in both settings, our models exhibit improved classification accuracy. Leveraging test-time scaling, we observe consistent gains as the number of sampled traces increases. Furthermore, we evaluate two open-source reasoning models, GPT-OSS-20B and Qwen3-14B, showing that while such models are capable of zero-shot reasoning, a lightweight approach—retraining only the embedding matrix of a frozen, smaller model like GPT-2 can surpass the performance of billion-parameter text-based reasoning models.

Index Terms— Audio Classification, Thinking, Reason

1. INTRODUCTION AND RELATED WORK

In recent years, Large Language Models (LLMs) have transformed artificial intelligence, tackling challenges once considered intractable, such as solving olympiad problems [1]. Their influence now extends beyond text, shaping natural language processing [2], acoustic modeling via tokens [3], raw audio generation [4], computer vision [5], and robotics [6]. A common paradigm is to discretize the modality into tokens and train a GPT-style model for next-token prediction, with reconstructions performed in the target domain if necessary from the discrete tokens. Beyond training, strategies such as test-time scaling [7, 8] have further advanced performance. For example, [8] demonstrated that verifying multiple sampled outputs of a LLM and selecting the best, optimal for a chosen criteria, improves the performance with the number of times we sample the output. In contrast with this, our work leverages LLMs to consume sequences of patch-level predictions ("reasoning trace") and directly output categories, without having explicit verification modules. Test-time scaling provides a powerful inference-time paradigm: the model and input remain fixed, while performance improves as the model "thinks longer" by generating and refining alternative solutions. In this paper, we adapt this framework to the domain of audio classification. Audio classification has historically followed advances in other domains, employing architectures like CNNs [9] and generative models [10]. Performance gains have traditionally been achieved by scaling model parameters or dataset sizes [11]. Specific to audio, another option has been to build a better front end keeping other parts of the model fixed [12, 13, 14] More recently, test-time scaling has emerged as a viable alternative, with methods like multi-level augmentations showing significant metric boosts in audio-captioning [15]. Extending this paradigm, we demonstrate that classification performance can be improved at test time even with a fixed model and input. We propose a method where, instead of naive input transformations, we aggregate evidence from stochastic audio patch predictions via reasoning model. We build a category trace that reflects the classifier's evolving hypothesis over time. Our LLM-based aggregation is inspired by Chain-of-Thought (CoT) prompting [16], which elicits intermediate reasoning steps in LLMs to improve performance, even in zero-shot setup [17]. This is enhanced through self-consistency over multiple reasoning chains [18]. While recent work explored CoT for audio from raw tokens [19], our approach complements this by constructing the reasoning chain from the outputs of a perceptual model. Here, the "steps" of reasoning correspond to evidence from different audio patches rather than textual exemplars. In summary, our contribution is a "thinking-while-listening" pipeline: at test time, we sample patch-level predictions from a frozen audio classifier to build a reasoning trace. A frozen LLM then leverages this trace to refine the final classification.

2. METHODOLOGY

We extend pre-trained audio models such as Audio Spectrogram Transformers (AST) [20] and YAMNet [21] to generate stepwise reasoning traces, and also explore new models de-

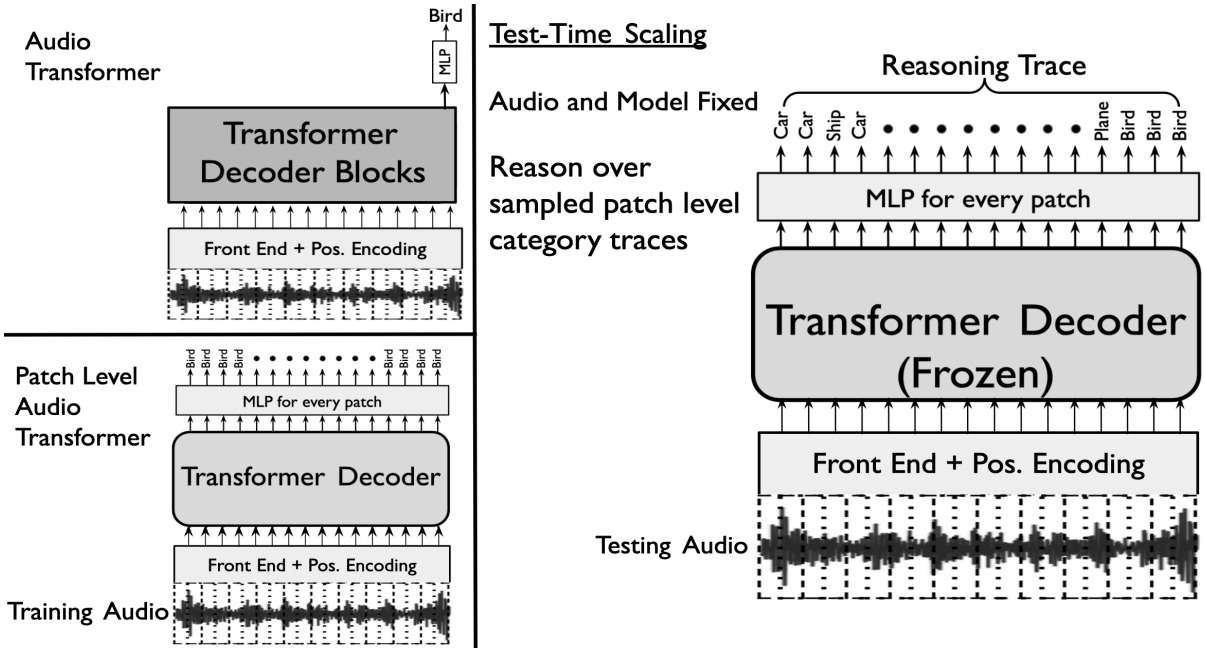

Fig. 1. Description of our proposed method. During training, we allow the model to causally predict categories for every patch to get the patch-level category output. The trained model is frozen during inference, and the audio is fixed. We sample from the posterior probability category distribution of each patch multiple times (which we define as the length of the sampling trace of a patch) to get a reasoning trace. The trace is then used to understand the category using a frozen LLM reasoning model like GPT-OSS 20B or a frozen GPT-2 model with a new embedding matrix to aggregate and give accurate predictions for the audio.

signed for this purpose. Motivated by advances in NLP, we adopt strategies from LLAMA [22] and GPT [23], where intermediate tokens (e.g., <think>, <wait>) elicit structured reasoning. Large-scale pretraining enables such tokens to invoke latent reasoning behaviors, which are leveraged through prompting after initial predictions. These reasoning prompts guide frozen models to refine outputs iteratively. Our goal is to construct analogous reasoning traces in audio tasks to enhance classification, with reasoning aided from text LLMs.

2.1. Reasoning Traces From Existing Classification Model

We show how audio understanding models developed as simple classification pipelines can be converted into models that can think and give us reasoning traces. In the context of this paper, the reasoning trace is a collection of category labels of what the model thinks the audio is until the waveform it has heard. The reasoning trace is a string of categories of length equal to the number of patches we break down the audio into casually, times the length of the sampling trace per patch, and is explained in the next subsection. For experiments with fixed 5s audio inputs in case of ESC-50 dataset, we vary the sampling trace length by segmenting audio into 500ms patches. In AST, predictions are generated causally after each new patch, concatenating them to increase the context for every output. As expected, initial predictions say the first 500ms are unreliable, yet performance improves steadily with longer context. This process yields a reasoning trace: at each patch, the sampling trace length corresponds to the number of times output probabilities are collected. Thus, a 5s audio produces $10 \times T$ category predictions, where T is the sampling trace length, each interleaved with posterior confidence. The full reasoning trace length is therefore $20 \times T$. Since the model is pretrained on AudioSet, categories are drawn from the AudioSet ontology [24]. While these may differ from the downstream dataset, semantic overlap allows the reasoning module to adapt effectively. Since models like YAMNet [21] and convolutional architectures such as VGG or ResNet [9] typically output a single embedding for a fixed audio input, adapting them to provide a reasoning trace is non-trivial. Similar to AST-style setups, for YAMNet we process audio in 500ms increments, generating predictions at each step. The convolutional encoder remains fixed, while predictions evolve with context length. Longer contexts yield more confident and accurate classifications. Interestingly, even short segments yield informative and reasonable predictions, often capturing fine-grained cues absent when processing full context as model tries hard to predict the best category in constrainted context.

2.2. Models That Can Think While Listening

Building on the initial results, we extend existing models such as Audio Transformers [25] to generate reasoning traces at test time, enabling a single forward pass over frozen audio and model inputs. As shown in Fig. 1, the baseline em-

ploys a single MLP head on the final Transformer decoder embedding, with AUC scores reported accordingly. Instead of training with a single target vector, category probabilities are predicted per patch during training using a sigmoid layer, with mean-squared error loss minimization. Since each 1s segment of FSD-50K may contain multiple categories, we modify the sampling process. Each MLP head which outputs a 200-dimensional vector is normalized to make it a probability distribution for patch-wise sampling. Both models are trained for 300 epochs with a learning rate of 1e-3, decaying to 1e-6.The front-end employs 64 filters and six layers of embedding dimension 64, similar to Audio Transformer [25].

2.3. Test Time Scaling Over Reasoning Traces

For both YAMNet (convolutional) and AST (transformer-based) models, audio is segmented into 500 ms chunks, and performance is evaluated as a function of sampling length. The per-chunk predictions form a reasoning trace that is aggregated into a final prediction, with majority voting as the simplest aggregator. In S1 [7], multiple outputs were combined using special tokens (e.g. <think>) to simulate step-wise reasoning, while for scaling methods using verifier aggregation is carried by sampling multiple candidates choose best optimized using verifier. In our setup, reasoning traces are generated causally: at each step, the model predicts likely categories conditioned on the audio observed. Candidates are sampled from the posterior distribution, where T defines the number of samples per patch, yielding a reasoning sequence of length $2P \times T$ for P patches, with confidence. This sequence is fed to reasoning models such as GPT-OSS 20B [26] and Qwen3 14B [27], guided by a structured prompt [1] to give the output category. This is a zero-shot setup, as models like S1, do not introduce additional parameters and keep the LLM decoder backbone frozen. We feed this trace to a frozen GPT-2 base model, re-training only the embedding matrix while keeping the weights and positional embeddings fixed, following the approach of Audio PALM [28]. The new embedding matrix vocabulary matches the number of problem categories, with 10 extra confidence tokens representing confidence scores (0–1) in 10 buckets, and the total output categories in the dataset of interest (50 for ESC-50, 200 for FSD-50K). Category tokens are interleaved with confidence tokens to form the reasoning trace, yielding $2PT$ tokens as input to GPT-2 (embedding size 768) across all experiments. This setup activates existing LLM connections while keeping the backbone fixed. Unlike [28], we pass the last token prediction through an MLP classification head with sigmoid activation [9] for multi-label outputs. Merely tuning the embedding matrix with frozen GPT-2 weights improves performance over open-source reasoning models (GPT-OSS4, Qwen3) using complex prompts. The performance improves with stronger reasoning models and increased length of posterior sampling for audio models.

[1] Prompt to GPT-OSS 20B and Qwen3 14B

"We take an audio wavform of 5 seconds and divide it into 10 patches each of 500ms. For each of the patch we sample multiple times and list the categories sampled from the distribution. For the entire audio waveform, can you predict which category the sound belongs to. For predicting the best category DO NOT COUNT or take the MEAN of the predicted categories. Rather reason through the category traces from patch 0 to 9 in a sequential manner. Reason and take into account what constitues a particular sound, what sub-atoms of a sound an audio is made of and draw the correlation from the category labels predicted to what best the sound patch and the entire trace progression would be. Take into account the confidence scores for each patch in the range of 0-100 with 100 being very confident and 0 being not at all confident for each of the patches. Here are the details of the audio file: The number of times each patch is sampled: 32.

CURRENT PATCH 0 – Categories for patch are: list of categories/conf.

CURRENT PATCH 1 – Categories for patch are: list of categories/conf.

So on ....

LIST OF CATEGORIES GIVEN

From the list please pick only one category most likely to be the audio

3. DATASET

We evaluate our framework on two widely used public audio classification datasets with available audio samples: ESC-50 [29] and FSD-50K [30], covering both single- and multi-label classification setups. The ESC-50 dataset consists of 50 audio categories with a total of 2,000 clips, each 5 seconds long and associated with a single label corresponding to the contents present in the audio. For this setup, we use backbone models built on pretrained networks, namely YAMNet and AST, and keep their weights frozen throughout training and inference. In contrast, FSD-50K involves predicting 200 categories from 1-second audio segments, where multiple labels may be present per clip. We adopt a consistent experimental protocol across both datasets: for ESC-50, we report top-1 accuracy, while for FSD-50K we evaluate performance using AUC for multi-label predictions. In both cases, we compare baseline model performance against results obtained with our test-time scaling approach, highlighting the improvements achieved by sampling techniques at inference.

4. EXPERIMENTS AND RESULTS

Our experiments highlight two consistent sources of improvement: (i) stronger pretrained reasoning models and (ii) longer patch-level sampling traces. We first evaluate frozen audio backbones on ESC-50, specifically YAMNet and AST trained on AudioSet with baseline as 84% on ESC-50 with model frozen. Full fine-tuning AST on ESC-50 yields 88.8% [20]. We can see that a frozen model can be pushed to a full-tuned performance with that of our proposed test-time scaling paradigm. As shown in Fig. 2, top-1 accuracy increases with trace length, with patch-level predictions fed into pretrained reasoning models including GPT-OSS 20B, Qwen3-14B, and frozen GPT-2 with a newly trained embedding matrix [23]. Despite frozen weights, AST consistently outperforms YAMNet, and both backbones benefit from test-time scaling. Table 2 indicates that GPT-OSS and Qwen do not surpass frozen GPT-2 with re-trained embedding matrix for this task.

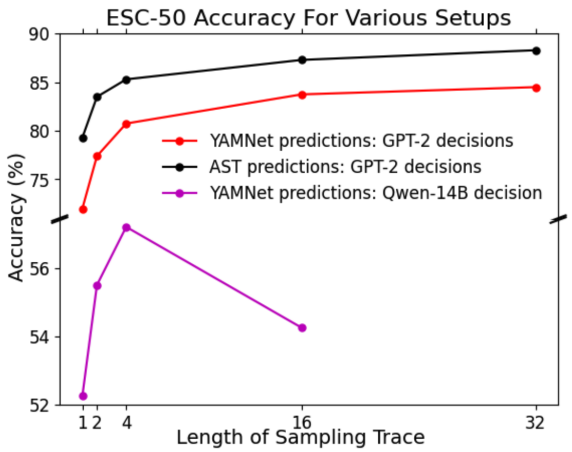

Fig. 2. Results for Test Time Scaling For ESC50 dataset for frozen YAMNet and AST as a function of length of sampling trace, using GPT-2 and Qwen3-14B for category prediction.

Table 1. ESC-50 accuracy with different sampling length for YAMNet backbone and zero shot text reasoning models.

Model	Sampling length/ output prediction			
	1	**2**	**4**	**16**
GPT-OSS 20B	53.5	58.75	57.6	**61.25**
Qwen3 14B	52.3	55.5	57.2	54.25

However, GPT-OSS designed for chain-of-thought reasoning—outperforms Qwen in our setup showing gains in reasoning performance carry forward to our task. Notably, all experiments keep audio backbones and transformer blocks of reasoning models fixed. Test-time performance improves with more posterior samples giving longer reasoning traces through repeated patch-level predictions. This establishes a robust test-time scaling paradigm, in contrast to standard audio classification pipelines, such as AST and YAMNet, which output a single probability vector. Adjusting temperature provides a minor gain (0.2%), insufficient to justify a large-scale study under the current setup. For our proposed architecture, we extend the framework to patch-level multi-label prediction by dividing each second into 40 patches similar to Audio Transformer baseline [25]. As shown in Fig. 3, shorter sampling traces do not improve over the baseline Audio Transformer AUC on 1-second inputs, likely due to information loss when sampling over single categories. Increasing the trace length ($1 \rightarrow 2 \rightarrow 8$ samples per patch) consistently boosts performance. When sampling at the patch level eight times per patch—our method surpasses the baseline. Across experiments, performance improves with longer traces and stronger reasoning models. Even weaker audio backbones, such as YAMNet, benefit from test-time scaling, demonstrating that model improvements are achievable regardless of initial backbone strength. Better architectures gives accurate reasoning traces, which in turn enhance test-time behavior.

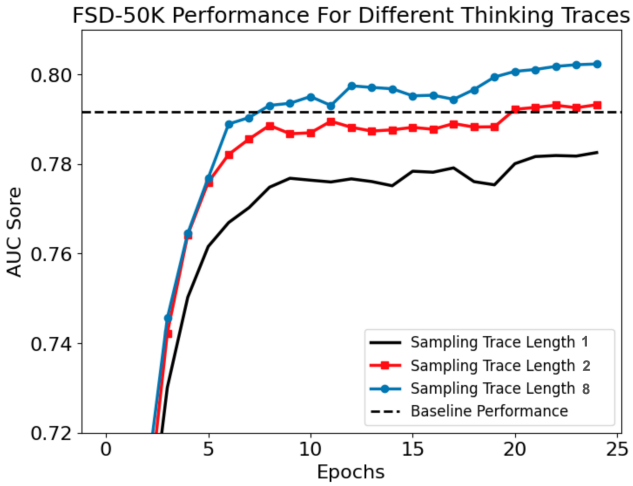

Fig. 3. Results on 1s chunks on FSD050K dataset for different length of sampling traces. We reason through the sampling trace with a frozen GPT-2 backbone, with trained embedding matrix with for baseline Audio Transformer, and backbone.

Table 2. ESC-50 accuracy with different temperatures/sampling trace lengths for AST backbone frozen, making predictions in 500ms increments using Frozen GPT-2.

Temp	Model	Sampling length / op prediction				
		1	**2**	**4**	**16**	**32**
1.0	YAMNet	72.0	77.4	80.8	83.8	84.5
1.0	AST	*Full Model Finetune* [20]				**88.8**
1.0	AST	79.3	83.5	86.3	87.3	**88.3**
1.2	AST	76.8	84.8	85.3	87.0	87.0
1.5	AST	72.5	80.5	82.8	86.5	**88.5**
2.0	AST	53.5	65.3	77.3	84.8	83.8

5. CONCLUSION AND FUTURE WORK

We present a simple approach for incorporating test-time scaling in audio classification, applicable to both existing pretrained models and a newly designed architecture that supports sampling of "reasoning traces" conditioned on an input audio signal. In addition, we improve baseline model by allowing multiple category predictions at each patch, which yields consistent gains as the sampling trace length increases. We provide ablation studies that integrate text-based reasoning and demonstrate that a frozen GPT-2, used for this task solely by retraining its embedding matrix, further improves performance. The proposed methods can be easily integrated into standard audio or image classification pipelines. Traditionally, advances in model performance have been attributed to increased scale—either in the number of parameters or the size of training data. In contrast, our results highlight a complementary direction: performance can be further scaled by reasoning over patch-level category traces during infer-

ence, while keeping both the backbone weights and input audio fixed. We observe this effect across two standard audio benchmarks, underscoring the generality of our approach. Our method is sufficiently general to enable test-time performance scaling for advancing a wide range of applications.

6. ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation (NSF) CAREER Award under Grant CCF-2236829, in part by the National Institutes of Health under Grant 1R01AG08950901A1, in part by the Office of Naval Research under Grant N00014-24-1-2164, and in part by the Defense Advanced Research Projects Agency under Grant HR00112490441.

7. REFERENCES

[1] Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong, "Solving olympiad geometry without human demonstrations," *Nature*, vol. 625, no. 7995, pp. 476–482, 2024.

[2] Tom B. Brown, Benjamin Mann, et al., "Language models are few-shot learners," 2020.

[3] Zalán Borsos, Raphaël Marinier, et al., "Audiolm: a language modeling approach to audio generation," 2023.

[4] Prateek Verma and Chris Chafe, "A generative model for raw audio using transformer architectures," in *2021 24th International Conference on Digital Audio Effects (DAFx)*. IEEE, 2021, pp. 230–237.

[5] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas, "Videogpt: Video generation using vq-vae and transformers," 2021.

[6] Anthony Brohan, Noah Brown, et al., "Rt-2: Vision-language-action models transfer web knowledge to robotic control," 2023.

[7] Niklas Muennighoff et al., "s1: Simple test-time scaling," *arXiv preprint arXiv:2501.19393*, 2025.

[8] Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini, "Large language monkeys: Scaling inference compute with repeated sampling," *arXiv preprint arXiv:2407.21787*, 2024.

[9] Shawn Hershey et al., "Cnn architectures for large-scale audio classification," in *ICASSP*, 2017.

[10] Prateek Verma and Julius Smith, "A framework for contrastive and generative learning of audio representations," *arXiv preprint arXiv:2010.11459*, 2020.

[11] Daniel PW Ellis and Nelson Morgan, "Size matters: An empirical study of neural network training for large vocabulary speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1999.

[12] Mirco Ravanelli and Yoshua Bengio, "Speaker recognition from raw waveform with sincnet," in *2018 IEEE spoken language technology workshop (SLT)*. IEEE, 2018, pp. 1021–1028.

[13] Neil Zeghidour, Olivier Teboul, Félix de Chaumont Quitry, and Marco Tagliasacchi, "Leaf: A learnable frontend for audio classification," in *International Conference on Learning Representations (ICLR)*, 2021.

[14] Prateek Verma and Chris Chafe, "A content adaptive learnable" time-frequency" representation for audio signal processing," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[15] Eungbeom Kim et al., "Exploring train and test-time augmentations for audio-language learning," *arXiv preprint arXiv:2210.17143*, 2022.

[16] Jason Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, 2022.

[17] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, 2022.

[18] Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch, "Frozen pretrained transformers as universal computation engines," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, vol. 36, pp. 7689–7697.

[19] Ziyang Ma, Zhuo Chen, Yuping Wang, Eng Siong Chng, and Xie Chen, "Audio-cot: Exploring chain-of-thought reasoning in large audio language model," *arXiv preprint arXiv:2501.07246*, 2025.

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

[21] Google, "Yamnet: A pre-trained audio event classifier based on audioset," `https://github.com/tensorflow/models/tree/master/research/audioset/YAMNet`, 2018.

[22] Hugo Touvron, Thibaut Lavril, et al., "Llama: Open and efficient foundation language models," 2023.

[23] Alec Radford et al., "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, pp. 9, 2019.

[24] Jort F. Gemmeke et al., "Audio set: An ontology and human-labeled dataset for audio events," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[25] Prateek Verma and Jonathan Berger, "Audio transformers: Transformer architectures for large scale audio understanding. adieu convolutions," *arXiv preprint arXiv:2105.00335*, 2021.

[26] OpenAI, "Gpt-oss-20b: Openai's open-weight mixture-of-experts reasoning model," *ArXiv preprint*, vol. abs/2508.10925, 2025, Model card / technical report for gpt-oss-20B.

[27] Jinze Bai et al., "Qwen-14b: A 14b-parameter transformer base model in the qwen series," Tech. Rep. Technical Report, Alibaba Group / Qwen Team, 2023.

[28] Paul K Rubenstein, Chulayuth Asawaroengchai, et al., "Audiopalm: A large language model that can speak and listen," *arXiv preprint arXiv:2306.12925*, 2023.

[29] Karol J. Piczak, "ESC: Dataset for environmental sound classification (esc-50)," in *Proceedings of the 23rd Annual ACM International Conference on Multimedia*. 2015, pp. 1015–1018, ACM.

[30] Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra, "FSD50K: an open dataset of human-labeled sound events," *arXiv preprint arXiv:2010.00475*, 2020.