Genre-Guided Music Transformation

Stavros Armeniakos and Fotis Bistas
School of Information Technology
Athens University of Economics and Business
Email: sta.armeniakos@aueb.gr, fot.bistas@aueb.gr

***Abstract*—This work explores novel techniques for genre-guided music transformation using a fine-tuned deep learning model. We begin by adapting Wav2Vec 2.0 [1], a model renowned for its robust audio representation capabilities, for the distinct task of music genre classification. This initial step leverages its pre-trained understanding of audio to accurately recognize genres such as rock, jazz, and hip-hop. Following the fine-tuning, we delve into the model's interpretability by analyzing the gradients of its predictions with respect to the input audio waveform. Crucially, we then employ a gradient-based optimization approach to subtly distort original audio samples. By iteratively adjusting the waveform in the direction suggested by the gradients, we can increase an audio sample's likelihood of being classified into a target genre, with a specific focus on transforming audio towards the "rock" genre. This innovative technique not only demonstrates a practical application of gradient-based input manipulation for controlled audio synthesis but also offers a powerful avenue for exploring genre transformation and enhancing interpretability within complex deep audio models.**

I. Introduction

Music genre classification is a foundational task in Music Information Retrieval (MIR) and has wide applications in music recommendation, archiving, and playlist generation. It involves developing systems that can automatically recognize the genre of an audio track, such as rock, jazz, classical, or hip-hop. Effective classification requires understanding complex and often subtle acoustic patterns that define each genre.

Traditionally, Convolutional Neural Networks (CNNs) [2] have been widely used in MIR due to their strong performance on spectrogram-based audio features. Recently, models like MERT [3] and MAEST [4] have advanced the state of the art by incorporating multi-modal or self-supervised learning techniques. Public datasets such as GTZAN [5] and Free Music Archive (FMA) [6] have played a critical role in benchmarking and evaluating the generalizability of these models.

These models typically operate on time-frequency representations of audio signals—most commonly, spectrograms. The spectrogram serves as a compact and expressive input format that captures both temporal and frequency-related patterns relevant to genre classification. Neural networks then process these representations to extract hierarchical features, which are subsequently mapped to a probability distribution over possible genres.

When working with audio, the spectral representation (like a spectrogram) often provides a rich and informative feature space. It highlights how different frequencies evolve over time, which is incredibly useful for analysis. However, this transformation from a raw time-amplitude waveform to a time-frequency representation isn't without its drawbacks, particularly when our goal is to reconstruct or modify the audio.

A significant challenge arises because the spectral representation discards crucial phase information from the original waveform. While the magnitude spectrum tells us "how much" of each frequency is present, the phase tells us "when" each frequency component peaks. This loss of phase makes it inherently difficult to perfectly reconstruct the original signal from its spectrogram alone, and more critically for our purposes, it complicates gradient-based modification methods. Without access to the complete phase, altering the signal in the spectral domain and then attempting to convert it back to a waveform can lead to inconsistent and unnatural-sounding results, as the phase information has to be estimated or discarded.

To circumvent these reconstruction and consistency issues, our approach directly utilizes the raw waveform as the input feature space for our model. This allows us to perform gradient-based manipulations directly on the time-domain signal, preserving all phase information. By operating on the raw waveform, we ensure that any modifications derived from the model's gradients can be consistently and accurately translated back into an audible audio signal, enabling precise and artifact-free genre-guided transformations.

In this work, we take a novel approach by applying gradient-based audio manipulation to explore how a fine-tuned Wav2Vec 2.0 [7] model can guide music audio to a target genre. Instead of generating music from scratch, we iteratively perturb an existing audio clip in the direction that maximizes its likelihood of classification as a desired genre. This technique serves as both an interpretability tool and a creative mechanism for blending and transformation of genres.

II. Related Work

Recent advances in audio representation learning have led to powerful models capable of extracting high-level features from raw waveforms. Among them, Wav2Vec [7] stands out as a general-purpose framework initially developed for speech. Trained using contrastive self-supervised learning, Wav2Vec 2.0 learns contextualized embeddings directly from unlabeled audio, enabling efficient fine-tuning on downstream tasks such as classification, tagging, and segmentation. Although designed with speech in mind, its architecture has been successfully adapted to broader audio applications, including music. In our work, we leverage Wav2Vec 2.0 as the backbone due to its ability to operate directly on waveforms and its accessibility

Code Available at: StavrosArm/genre-guided-music-transformation

as an open-source, pre-trained model. Compared to larger or more domain-specific alternatives, it offers a practical balance between performance and resource efficiency—making it well suited for genre classification under constrained computational budgets.

Building on this trend of adaptable and efficient models, Alonso-Jiménez et al. introduce MAEST (Music Audio Efficient Spectrogram Transformer), a convolution-free transformer architecture trained in a fully supervised manner for music representation learning [4]. MAEST builds on ideas from AST [8] and PaSST [9], and uses a large, Discogs-derived dataset [10] combined with patchout regularization at both training and inference time. This allows it to process long input sequences (up to 30 seconds) while keeping GPU usage manageable.

Complementary to supervised efforts, Li et al. present MERT (Music undERstanding Transformer), a scalable self-supervised framework that combines timbre and pitch cues via dual teacher signals—a Residual VQ-VAE (EnCodec) and a Constant-Q spectrogram—within a masked prediction setup [3]. Trained on up to 160 kHz of music, MERT scales from 95M to 330M parameters while remaining trainable on a single GPU thanks to short 5-second crops and architectural optimizations like attention relaxation and Pre-LN. Without task-specific fine-tuning, MERT matches or exceeds the performance of larger baselines like Jukebox [11] across 14 MIR tasks, making it a highly efficient and generalizable music representation model.

Beyond representation learning, our work also draws inspiration from a different but related area: adversarial attacks. Originally studied in image classification, adversarial examples—introduced by Szegedy et al. [12] and expanded with methods like FGSM [13] and the C&W attack [14]—demonstrated how small, imperceptible perturbations could cause high-confidence misclassifications. These ideas were later extended to the audio domain. For example, Carlini and Wagner [15] crafted targeted audio examples that sound benign to humans but produce adversarial transcriptions in ASR systems. Subsequent work by Qin et al. [16] and Yakura and Sakuma [17] made such attacks more robust and applicable even in physical playback scenarios.

Inspired by these techniques, we repurpose the adversarial attack framework—not to deceive a classifier—but to guide a genre transformation process. By computing input perturbations through gradient-based optimization, we aim to shift an audio clip's genre embedding in a targeted and semantically meaningful direction, enabling controllable transformations without retraining the model. This approach bridges ideas from representation learning and adversarial inference, offering a novel lens on genre manipulation in the musical domain.

III. Methodology

A. EDA (Exploratory Data Analysis)

For our purposes, we use the well-known FMA-Small dataset [6], a curated subset of the Free Music Archive containing 8,000 songs evenly distributed across 8 balanced genres.

We performed a number of data preprocessing and analysis tasks, genre metadata analysis, and structured representation of musical features to facilitate genre-guided transformation. The workflow is composed of several stages, each implemented in a dedicated processing notebook.

We computed the average duration of tracks across different genres to inform our selection of input data and optimize training sequence lengths. This analysis enabled us to prioritize genres with durations close to the overall mean, avoiding those with significantly longer or shorter average lengths that could introduce variability or bias into the model.

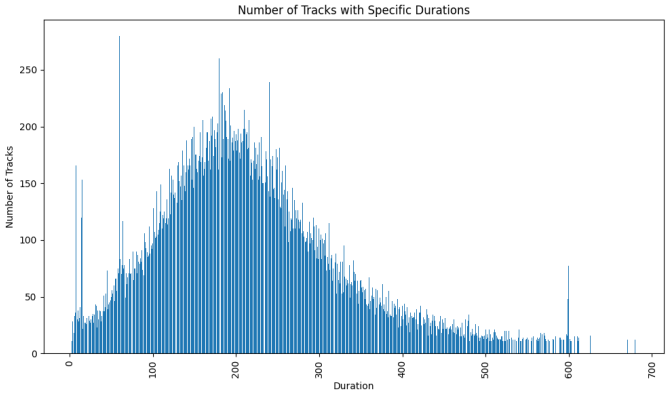

Fig. 1. Distribution of track durations in the dataset, showing a peak around 200 seconds and a long-tail decline for longer tracks.

It can be observed that the average song for the dataset is around 2 minutes. Using the average duration per genre we can see which genres fall around this average.

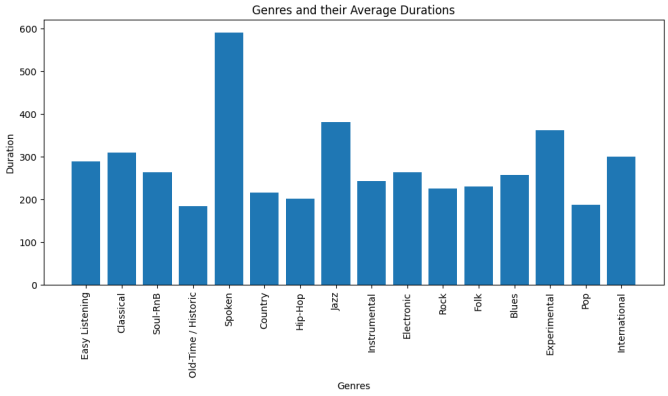

Fig. 2. Average track duration per genre, highlighting significant variation, with genres like Spoken and Experimental showing notably higher means.

We also analyzed the frequency with which different genres co-occur across individual tracks. This co-occurrence matrix reveals relationships and overlap between genres, such as common pairings (e.g., Jazz and Blues) or rare combinations. By quantifying these associations, we gain insights into genre similarity. We know for example that if Blues and Jazz are

closely related, generating one from the other should be easier than generating two unrelated genres.

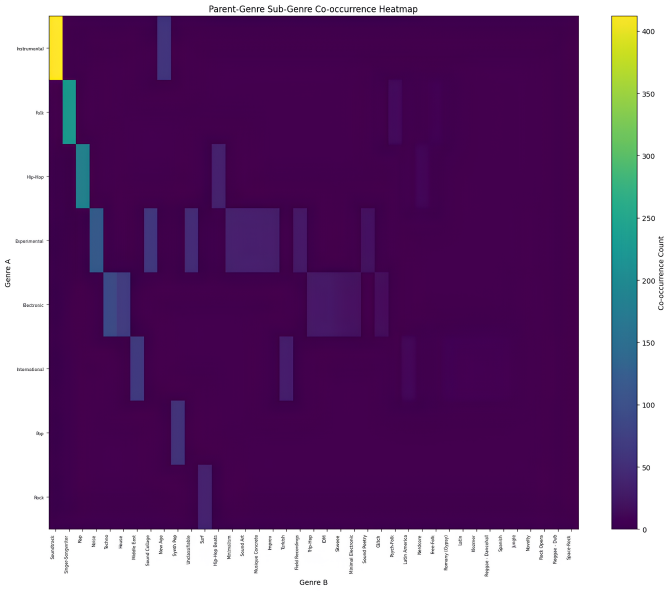

Fig. 3. Heatmap of genre co-occurrence frequencies, showing how often sub-genres appear together across tracks. Brighter regions indicate stronger associations, which can guide genre embedding design.

B. Genre classification

Genre classification accepts an audio file path and returns both a discrete label and its numerical index. First, the raw waveform is loaded via torchaudio and passed through a preprocessing pipeline.

The audio preprocessing pipeline operates in five sequential stages to ensure consistency and robustness across inputs. First, multi-channel signals are collapsed to a single channel by averaging, yielding a mono waveform. Second, if the original sampling rate differs from 16 kHz, the signal is resampled via a sinc-based bandlimited filter to the target rate. Third, a fixed time interval—specified in 16 kHz samples—is extracted, after which the segment is either zero-padded or truncated to a uniform length of 10s (160 000 samples). Fourth, optional additive Gaussian noise, scaled by a user-defined ratio, is injected to simulate environmental variability. Finally, the output is clamped to the $[-1, 1]$ amplitude range, producing a normalized, fixed-length tensor ready for downstream feature extraction or model inference.

The model is built upon a pre-trained self-supervised encoder (Wav2Vec 2.0), which remains frozen during training. A lightweight classification head is added on top, and linear fine-tuning is performed—i.e., only the classifier parameters are optimized while the encoder weights are kept fixed. This setup allows for adaptation to the genre classification task under limited computational resources.

The forward pass produces a vector of class-conditional scores, from which the maximum entry is selected. Finally, this index is mapped to its corresponding genre string (e.g., "rock," "jazz," "hip-hop") via a static lookup table. The model was trained for approximately 10 hours on an Nvidia G6 xlarge.

Training Configuration:

- **Model:** `facebook/wav2vec2-base` (frozen encoder)
  - First hidden layer: 768x515
  - Classification head: 512
  - Number of labels: 8
  - GeLU activation function
- **Optimizer:** AdamW
  - Learning rate: 5e-4
  - Weight decay: 1e-5
  - Scheduler: StepLR (gamma = 0.5, step every 5 epochs)
- **Training Setup:**
  - Batch size: 16
  - Epochs: 20
  - Dropout probability: 0.4
  - Early stopping patience: 5
  - Random seed: 24

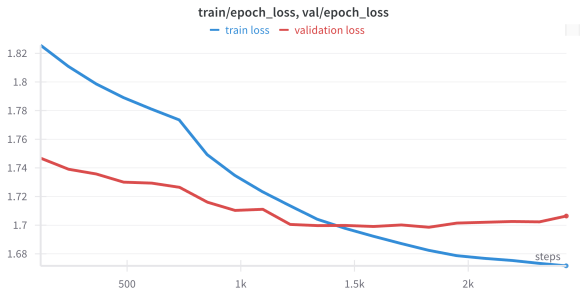

Fig. 4. Training and validation loss over epochs.

Analysis: The training loss (blue) steadily decreases, while the validation loss (red) initially improves but begins to plateau and slightly rise after approximately 1500 steps, suggesting the onset of overfitting. This indicates that the model is starting to learn patterns specific to the training data that do not generalize well. However, the relatively small gap between the training and validation losses, along with the overall high loss values, suggests that the model is also underfitting to some extent. This apparent contradiction points to a limitation not in the training process per se, but in the expressiveness of the classification head itself—which may lack the capacity to capture the full complexity of the underlying data manifold.

Below, in Figure 5 and 6 we present evaluation metrics on the validation set and the held-out test set, which further support the underfitting hypothesis. Despite a well-regularized training setup, the model fails to achieve strong generalization performance, indicating that it lacks sufficient representational flexibility. The next logical step in model development would be to progressively unfreeze layers of the encoder during training—starting with the later (task-relevant) layers. However, this approach was not feasible within the constraints of

the available computational resources, and remains an open avenue for future work.

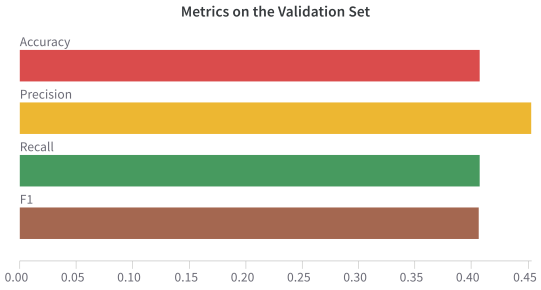

Fig. 5. Metrics on the validation set.

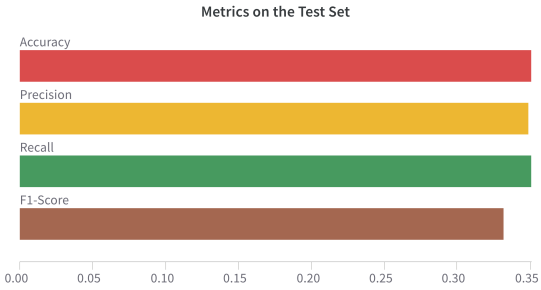

Fig. 6. Metrics on the test set.

C. Genre-Guided Music Transformation

To explore genre transformation using a discriminative model, we leverage the aforementioned fine-tuned Wav2Vec 2.0 classifier to estimate the divergence between a given input audio waveform and a target genre, specifically rock. Central to our approach is the use of the gradient of the loss function with respect to the input waveform, $\nabla_{\hat{\mathbf{x}}}\mathcal{L}$, which provides a principled direction for modifying the input. This gradient indicates how each element of the waveform should be altered to reduce the loss and thereby increase the model's confidence in assigning the desired target label.

By iteratively updating the waveform in the direction of this gradient, we gradually reshape the input so that its internal representation becomes more aligned with that of typical examples from the rock genre. In doing so, we preserve perceptual similarity to the original while steering the model's prediction toward the target class. This process effectively uses the model's learned decision boundaries as a guide for semantically meaningful, genre-specific audio transformation.

We define an objective function that balances two components:

- Cross Entropy loss: Measures how close the predicted distribution of the model is to a one-hot target distribution representing the rock genre.
- l_2-loss: Penalizes deviation from the original waveform to preserve perceptual similarity.

Given the input waveform x, the distorted waveform $\hat{x}$, the model $f(\cdot)$ and the hyperparameter λ, we formulate the aforementioned task as the following optimization problem:

$$\begin{aligned} \min_{\hat{\mathbf{x}}} \quad & -\sum_i \mathbf{y}_{\text{rock},i} \cdot \log\left(f_i(\hat{\mathbf{x}})\right) + \lambda \cdot \|\hat{\mathbf{x}} - \mathbf{x}\|_2^2 \\ \text{s.t.} \quad & \hat{\mathbf{x}} \in [-1, 1] \end{aligned}$$

The constraint ensures that the audio waveform remains within a valid amplitude range for playback and storage. After each gradient update, values of $\hat{\mathbf{x}}$ are clipped to $[-1, 1]$ to prevent instability and artifacts.

The gradient of the loss function is as follows:

$$\mathcal{L}(\hat{\mathbf{x}}) = -\sum_i \mathbf{y}_{\text{rock},i} \cdot \log\left(f_i(\hat{\mathbf{x}})\right) + \lambda \|\hat{\mathbf{x}} - \mathbf{x}\|_2^2$$

$$\nabla_{\hat{\mathbf{x}}}\mathcal{L} = -\sum_i \left(\frac{\mathbf{y}_{\text{rock},i}}{f_i(\hat{\mathbf{x}})} \cdot \nabla_{\hat{\mathbf{x}}} f_i(\hat{\mathbf{x}}) \right) + 2\lambda(\hat{\mathbf{x}} - \mathbf{x})$$

We iteratively compute the gradient of the input signal with respect to the loss function until convergence. In Algorithm 1 we provide pseudo-code of our implementation.

Algorithm 1 Genre Transformation via Gradient-Based Optimization (Cross-Entropy)

$\hat{\mathbf{x}} \leftarrow \mathbf{x}$
while true **do**
  $\mathbf{p} \leftarrow \text{softmax}(f(\hat{\mathbf{x}}))$
  $\mathcal{L}_{\text{CE}} \leftarrow -\sum_i \mathbf{y}_{\text{rock}}[i] \cdot \log\left(\mathbf{p}[i]\right)$
  $\mathcal{L}_{\text{prox}} \leftarrow \|\hat{\mathbf{x}} - \mathbf{x}\|_2^2$
  $\mathcal{L} \leftarrow \mathcal{L}_{\text{CE}} + \lambda \cdot \mathcal{L}_{\text{prox}}$
  if $\mathcal{L} < \epsilon$ **then**
    break
  end if
  Compute gradients $\nabla_{\hat{\mathbf{x}}}\mathcal{L}$
  $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} - \eta \cdot \nabla_{\hat{\mathbf{x}}}\mathcal{L}$
  $\hat{\mathbf{x}} \leftarrow \text{clip}(\hat{\mathbf{x}}, -1, 1)$
end while
return $\hat{\mathbf{x}}$

1) Fréchet Audio Distance (FAD) [18]: To objectively assess the perceptual realism of our genre-transformed audio, we utilize the **Fréchet Audio Distance (FAD)** [18]. FAD is a prominent metric for evaluating the quality and diversity of generated audio by comparing the statistical properties of its feature distribution against that of a reference dataset of real audio. It measures the Fréchet distance between two multivariate Gaussian distributions.

Given two sets of audio embeddings, $\mathbf{X}_1$ representing real audio and $\mathbf{X}_2$ representing generated audio, with their respective mean vectors $(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)$ and covariance matrices $(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2)$, the FAD is formally defined as:

$$FAD = \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2^2 + \text{Tr}(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2 - 2(\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_2)^{1/2}) \quad (1)$$

A lower FAD score indicates that the distribution of generated audio embeddings is perceptually more similar and realistic, aligning more closely with the characteristics of genuine audio. For our task, we compute FAD between the embeddings of our generated, rock-transformed audio samples and the embeddings of a diverse collection of authentic rock music. This metric provides a quantitative complement to qualitative listening evaluations, ensuring our transformations yield high-fidelity and genre-consistent results.

TABLE I
FAD SCORES FOR GENRE-TRANSFORMED AUDIO.

Original Genre	Target Genre	FAD Score
Pop	Rock	1887.8
Folk	Rock	1713.2
Electronic	Rock	1817.02

IV. CONCLUSION

In this work, we explored genre-guided transformation of music audio using self-supervised representations and gradient-based perturbations. While the core objective of achieving reliable genre control was hindered by the limited capacity of the classifier—constrained in part by computational resources—we were still able to produce interesting audio results. These results suggest that the underlying methodology holds promise, even if full genre disentanglement was not achieved under current settings.

Despite the limitations, our findings open up a rich avenue for future research at the intersection of self-supervised audio representation learning and controllable music transformation, where subtle and interpretable modifications to musical attributes remain a compelling and largely untapped frontier.

REFERENCES

[1] Y. Li, R. Yuan, G. Zhang, Y. Ma, C. Lin, X. Chen, A. Ragni, H. Yin, Z. Hu, H. He, E. Benetos, N. Gyenge, R. Liu, and J. Fu, "Map-music2vec: A simple and effective baseline for self-supervised music audio representation learning," 2022. [Online]. Available: https://arxiv.org/abs/2212.02508

[2] J. Pons and X. Serra, "musicnn: Pre-trained convolutional neural networks for music audio tagging," 2019. [Online]. Available: https://arxiv.org/abs/1909.06654

[3] Y. Li, R. Yuan, G. Zhang, Y. Ma, X. Chen, H. Yin, C. Xiao, C. Lin, A. Ragni, E. Benetos, N. Gyenge, R. Dannenberg, R. Liu, W. Chen, G. Xia, Y. Shi, W. Huang, Z. Wang, Y. Guo, and J. Fu, "Mert: Acoustic music understanding model with large-scale self-supervised training," 2024. [Online]. Available: https://arxiv.org/abs/2306.00107

[4] P. Alonso-Jiménez, X. Serra, and D. Bogdanov, "Efficient supervised training of audio transformers for music representation learning," 2023. [Online]. Available: https://arxiv.org/abs/2309.16418

[5] B. L. Sturm, "The state of the art ten years after a state of the art: Future research in music information retrieval," *Journal of New Music Research*, vol. 43, no. 2, pp. 147–172, Apr. 2014. [Online]. Available: http://dx.doi.org/10.1080/09298215.2014.894533

[6] K. Benzi, M. Defferrard, P. Vandergheynst, and X. Bresson, "FMA: a dataset for music analysis," *CoRR*, vol. abs/1612.01840, 2016. [Online]. Available: http://arxiv.org/abs/1612.01840

[7] 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 (NeurIPS)*, 2020.

[8] Y. Gong, Y.-A. Chung, and J. Glass, "Ast: Audio spectrogram transformer," 2021. [Online]. Available: https://arxiv.org/abs/2104.01778

[9] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," in *Interspeech 2022*, 2022. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2022-227

[10] R. O. Araz, X. Serra, and D. Bogdanov, "Discogs-vi: A musical version identification dataset based on public editorial metadata," 2024. [Online]. Available: https://arxiv.org/abs/2410.17400

[11] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," 2020. [Online]. Available: https://arxiv.org/abs/2005.00341

[12] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[13] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[14] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," *IEEE Symposium on Security and Privacy*, 2017.

[15] ——, "Audio adversarial examples: Targeted attacks on speech-to-text," *2018 IEEE Security and Privacy Workshops (SPW)*, 2018.

[16] Y. Qin, N. Carlini, G. Cottrell, I. Goodfellow, and C. Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," *arXiv preprint arXiv:1903.10346*, 2019.

[17] H. Yakura and J. Sakuma, "Robust audio adversarial example for a physical attack," *arXiv preprint arXiv:1810.11793*, 2019.

[18] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fréchet audio distance: A metric for evaluating music enhancement algorithms," 2019. [Online]. Available: https://arxiv.org/abs/1812.08466