HarmonyCloak: Making Music Unlearnable for Generative AI

Syed Irfan Ali Meerza*, Lichao Sun†, Jian Liu*
*University of Tennessee, Knoxville, TN, USA
†Lehigh University, PA, USA
Emails: smeerza@vols.utk.edu, lis221@lehigh.edu, jliu@utk.edu

Abstract—**Recent advances in generative AI have significantly expanded into the realms of art and music. This development has opened up a vast realm of possibilities, pushing the boundaries of human creativity into unexplored frontiers. However, as generative AI advances, it can replicate artistic styles and produce new artwork, posing significant concerns for the perceived rarity and value of artists' creations. In response to these challenges, it is becoming increasingly crucial to establish and enforce protective measures that safeguard artists' copyrighted work from unauthorized exploitation by generative AI models. In this paper, we introduce the first defensive mechanism, HarmonyCloak, to prevent the exploitative use of artwork, specifically in the context of instrumental music, by generative AI models. Particularly, HarmonyCloak employs imperceptible *error-minimizing* noise to make the model's generative loss approach zero for these perturbed music data, tricking the model into believing nothing can be learned so as to disrupt their attempts to replicate musical structures and styles. By using a set of intra-track and inter-track objective metrics and a subjective user study, extensive experiments on three state-of-the-art music generative AI models (i.e., MuseGAN, SymphonyNet, and MusicLM) validate the effectiveness and applicability of HarmonyCloak[1] in both white-box and black-box settings.**

1. Introduction

In recent years, the world has witnessed the remarkable rise of generative AI in various fields [1]. From the generation of high-quality and hyper-realistic images (e.g., DALL-E-2 [2]) to text generation capable of coherent and contextually relevant writing (e.g., ChatGPT [3]), generative models have revolutionized various domains. These advancements have also made significant strides in music and audio generation, enabling the creation of original compositions and elevating the quality of audio experiences [4], [5]. Notably, AI was instrumental in completing Beethoven's unfinished "Tenth Symphony" for his 250th-anniversary celebration in Bonn [6], and in creating "Tokyo 2020 Beats" for the Tokyo Olympics [7]. These notable achievements underscore the profound and far-reaching influence of generative AI.

However, the rise of music-generative AI brings a significant concern regarding the unauthorized exploitation of musicians' composed or copyrighted music [8]. As generative AI algorithms learn from vast databases of music, there is a risk that they may inadvertently generate compositions that bear striking similarities to existing works, potentially infringing upon copyright protections. This unauthorized usage of musicians' intellectual property could have severe consequences, negatively impacting the livelihoods and creative rights of these artists. Not only can it result in financial losses for musicians, but it also undermines their artistic integrity and hampers their ability to control the distribution and use of their original compositions [9]. Without appropriate safeguards and legal frameworks in place, the exploitation of musicians' work through generative AI poses a significant threat to the vibrant and diverse ecosystem of music creation and the artists who contribute to it.

Prior Research in AI Data Protection. In recent years, several efforts (e.g., [10], [11], [12], [13]) have been made to protect data from unauthorized usage by making data samples *unlearnable*. This involves introducing imperceptible *error-minimizing* noise into the data, effectively degrading the performance of models trained on the perturbed data. Unlike conventional data protection techniques, such as differential privacy [14] and data-encrypting approaches [15], [16], *unlearnable examples* (UEs) do not compromise the quality of data for normal usage while remaining unexploitable to AI models. Despite their effectiveness, mainly in image classification, UEs have not been explored much outside this domain, such as audio. Moreover, these approaches often assume a white-box setting where full access to model parameters and architecture is available, limiting their applicability in real-world scenarios. Another line of research [10], [17] employs *distance-maximizing* noise to substantially shift the image examples within the feature space, aiming to hinder unauthorized use. However, this approach does not render the data unlearnable; instead, it forces the model to learn incorrect representations, adversely affecting its overall performance.

Challenges. Several unique challenges arise when it comes to generating UEs for music-generative AI models: (1) The inherent nature of generative models poses a difficulty as they aim to learn and replicate complex patterns, structures, and styles of existing musical compositions. Unlike classification models, generative AI models operate

[1] Audio examples of the unlearnable music examples are available for listening at https://mosis.eecs.utk.edu/harmonycloak.html.

fundamentally differently, making it necessary to develop specialized techniques tailored to the characteristics of generative models; (2) The lack of a clear ground truth for unlearnability in the context of generative models exacerbates the challenge. Unlike in classification tasks where the target is well-defined, assessing the unlearnability of generative models becomes more nuanced and subjective; and (3) The time-sensitive and complex nature of music, with evolving melodies and rhythms, makes generating unlearnable music while preserving its essence a significant challenge.

HarmonyCloak. In response to these challenges, this paper proposes HarmonyCloak, the first defensive framework to render unlearnable examples in music, aiming to assist musicians and the music industry in safeguarding their content against unauthorized use by generative AI technologies. In this work, we focus primarily on instrumental music with varied rhythmic structures and instrumentation, given the limited availability of open-source generative models for vocal music. Specifically, we strategically incorporate imperceptible *error-minimizing* noise into the music training samples, effectively minimizing the generative loss for these perturbed samples to trick the model into believing nothing can be learned. This incorporation of noise serves as a protective measure to prevent the informative knowledge of the data from being learned or extracted. Importantly, by ensuring that the modifications remain imperceptible to the human ear through a set of *time-dependent* optimization constraints, we can maintain the original essence and artistic integrity of the music, allowing it to be enjoyed without compromising its perceptual quality. Our approach also accounts for the music production process, acknowledging that musicians may convert their compositions into lossy compression formats (e.g., MP3) for distribution. To enhance the practicality and versatility of the proposed framework, we have also devised tailored approaches for both white-box and black-box settings. Our major contributions can be summarized as:

- To the best of our knowledge, HarmonyCloak is the first defensive framework to make instrumental music unlearnable for generative AI models. Our approach involves adding imperceptible *error-minimizing* noise to the music, effectively making it unlearnable without compromising its perceptual quality.
- Our framework goes beyond traditional L_p-norm-based or psychoacoustic-hiding-based methods by strategically crafting imperceptible noise leveraging both the human hearing threshold and dynamic time-dependent musical characteristics, ensuring minimal perceptual impact while enhancing effectiveness.
- To ensure the versatility of our method, we have devised tailored approaches for both white-box and black-box settings to effectively generate imperceptible noises, ensuring that our method can be applied in a wide range of practical scenarios.
- Extensive experiments on three state-of-the-art music generative models, MuseGAN [18], SymphonyNet [19], and MusicLM [20], across various experimental settings and practical scenarios demonstrate the effectiveness and applicability of HarmonyCloak.

TABLE 1: Comparison of noise-based attacks/defenses.

Method	Init. Phase	Noise Type	Objective
Adversarial Attacks	Testing	Error-maximizing	Degrade model performance
Adversarial Training	Training	Error-maximizing (*Min-max*)	Strengthen model robustness
Data Poisoning Attacks	Training	Trigger or Error-maximizing	Degrade model performance
Unlearnable Examples	**Training**	**Error-minimizing (*Min-min*)**	**Make data unexploitable**

2. Preliminaries

2.1. Unlearnable Examples

For a model to effectively learn, there must be a discernible knowledge gap between its current understanding and the new data it encounters, which is quantified by the loss generated from each data sample. Conventionally, the concept of crafting UE(s) for classification models are based on how to modify each data sample so that the classification loss remains close to zero. This *zero loss* tricks the model into believing there is "nothing" to learn from these example(s) [11]. Explicitly, given a data input x with label y, the defender can generate *error-minimizing* noise δ by solving the following bi-level optimization problem:

$$\arg\min_{\theta} \quad \mathbb{E}_{x,y}[\min_{\delta}\mathcal{L}(f(x+\delta),y)] \quad s.t. \quad \|\delta\|_p \leq \epsilon, \quad (1)$$

where f denotes the model, $\mathcal{L}$ is the cross-entropy loss, and the noise magnitude is bounded by $\|\delta\|_p$. However, unlike classification models, generative models operate in a fundamentally distinct manner, making it challenging to directly utilize this approach for generating UEs.

Comparison of Noise-based Attacks/Defenses. Table 1 shows a comparison of various methods (attacks or defenses) in AI that rely on noise injection into data. Adversarial attacks [21], [22], [23] aim to maximize AI models' prediction errors by injecting *error-maximizing* noise during the testing phase. Adversarial training [24], [25] seeks to enhance model robustness by integrating adversarial examples into the training phase, which can be formulated as a *min-max* optimization problem. Data poisoning attacks [26] degrade the model's performance by tampering with its training data. Backdoor attacks [27], [28], as a special case of poisoning attacks, embed stealthy triggers in the training data, misleading the model to incorrectly respond to data containing trigger patterns. Research [29] has also shown that the use of *error-maximizing* noise for data poisoning is highly effective. However, applying *error-maximizing* noise to training samples will not stop the model from learning. Conversely, unlearnable examples take an opposite approach, i.e., injecting *error-minimizing* noises through a *min-min* optimization process, to trick the model into believing that nothing can be learned from these examples, effectively rendering the data unexploitable.

2.2. Representation of Music Signals

Polyphonic music data, i.e., the Musical Instrument Digital Interface (MIDI) format [30], has long been a cornerstone in music composition and editing. Unlike traditional audio files, MIDI does not contain actual sound but

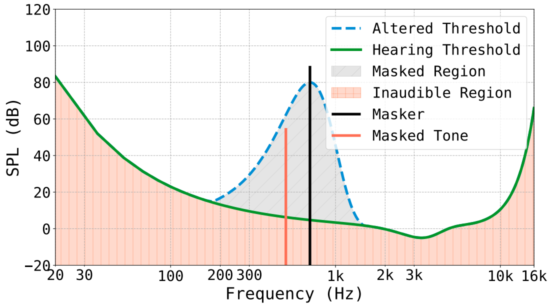


Figure 1: Illustration of frequency masking in psychoacoustic modeling (log-scaled x-axis: 20 Hz — 16 kHz).

rather provides detailed instructions on how music should be played, specifying notes, their lengths, and intensity. Today, most composers use digital audio workstations (DAWs) (e.g., Ableton Live, Logic Pro, and FL Studio), which heavily rely on MIDI for composing and arranging music [31]. On the other side, some recent music generative models, such as Google's MuLan [32] and MusicLM [20] illustrate the trend towards using audio data directly, rather than MIDI. The move towards using direct audio formats allows for a more nuanced and high-fidelity generation of music, capturing the subtleties of timbre and expression that MIDI might not fully replicate. **In light of these developments, HarmonyCloak needs to accommodate both MIDI and raw audio formats, such as *.wav* and *.mp3*, to address the evolving landscape of music generation.**

The MIDI format can be visualized as a matrix akin to a musical scoresheet, with values ranging from 0 to 1 (i.e., *note velocity*), where each element represents the presence or absence of notes at different time steps. To be precise, a multi-track piano-roll for M instrumental tracks within a single bar can be represented as a tensor $x \in [0, 1]^{R\times S\times M}$, with R denoting the number of time steps in a bar and S indicating the number of available note candidates. The raw audio waveform format, which can be either converted from the MIDI files using DAWs or recorded from physical instruments, can be represented as $x \in [-1, 1]^N$, where N is the number of samples.

2.3. Particular Aspects of Music

While there has been active research on audio machine learning attacks relying on imperceptible noise injection (e.g., [33], [34], [35], [36]), they primarily target the manipulation of speech data to deceive speech recognition or speaker identification systems. These methods employ either the L_p norm or psychoacoustic hiding approaches [37], [38] to constrain the added noise, ensuring it remains imperceptible to listeners. However, these approaches might not be directly applied to music data as it encompass a broader range of frequencies, more complex harmonic structures with multiple instrumental tracks, and dynamic variations in tempo and volume that are not present in speech.

Moreover, music often involves multiple instruments and voices simultaneously, adding layers of complexity to its acoustic and perceptual properties. A recent study [39] proposes a human-in-the-loop attack to create adversarial music to evade copyright detection. However, this method requires considerable human-involved attacking effort and may vary in effectiveness due to the complexity of music perception and the diverse ways in which individuals experience and interpret music [40], [41]. Additionally, none of the existing research has considered the music production process, where musicians often convert their compositions into specific formats (e.g., MP3) for distribution. This conversion typically includes lossy compression, potentially affecting the efficacy of the introduced noises. **Therefore, ensuring the perceptual quality of music when injecting noise demands a more nuanced and low-effort approach that accounts for both the complex nature of compositions and the music production process.**

2.4. Psychoacoustic Modeling

Psychoacoustic models [42] have revealed that sounds falling below certain psychoacoustic hearing thresholds, i.e., the absolute hearing threshold, become imperceptible to humans. This hearing threshold, as shown in Figure 1, can be calculated for any frequency (ν) as:

$$T_H(\nu) = 3.64 \times \left(\frac{\nu}{1000}\right)^{-0.8} - 6.5 \times e^{-0.6\times\left(\frac{\nu}{1000}-3.3\right)^2}. \quad (2)$$

Moreover, the physical apparatus in the ear used to detect sound can be overwhelmed by the louder sound, raising the hearing threshold for other sounds at nearby frequencies, a phenomenon referred to as *"frequency masking"* [43]. When measuring frequency masking curves, researchers discovered a *critical bandwidth* around the masker frequency where the masking threshold remains flat rather than dropping off. In Figure 1, we can observe that the critical bandwidth forms a bell curve in which the hearing threshold is elevated in the presence of a loud masker tone. Despite having a higher level than the hearing threshold, tones falling within this curve will be masked out by the masker. Additionally, masking can occur even when the masker and maskee sounds are not played simultaneously, a phenomenon known as *"temporal masking"* [43]. Louder sounds can obscure quieter ones immediately before (premasking) or after (post-masking) their occurrence, further highlighting the temporal dynamics of auditory perception.

MP3 Lossy Compression. Leveraging insights from psychoacoustic models, audio compression technologies like MP3 [44] can strike an optimal balance between audio quality and file size, making them ideal for music distribution. Specifically, MP3 compression leverages frequency masking by calculating the masking-to-noise ratio (MNR), enabling the encoder to allocate fewer bits for masked regions and effectively compress the audio to a smaller file size. Given that composers may choose MP3 compression for music distribution due to its wide compatibility and compactness, it is important to make sure that any defensive noises introduced by HarmonyCloak are robust enough to withstand this lossy compression process.

TABLE 2: Target music generative AI models.

Model	Year	Backbone	Type of Music	#Params
MuseGAN [18]	2018	GAN	Multi-track MIDI	720k
SymphonyNet [19]	2022	Transformer	Multi-track MIDI	30M
MusicLM [20]	2023	Transformer	Audio Waveform	1B

2.5. Deep Music Generation

While early attempts at music generation mainly use the recurrent neural network (RNN) based architectures [45], [46], [47], more recent advancements in deep music generation, since 2017, have leveraged more powerful deep generative models, such as Generative Adversarial Networks (GAN) and Transformers. For instance, MidiNet [48] and MuseGAN [18] utilize CNN-based GAN architectures for music generation. SeqGAN [49] employs an RNN-based GAN framework, where the generative model is modeled as a stochastic policy in reinforcement learning. Recently, due to the exceptional effectiveness of Transformer models in natural language processing (NLP) and computer vision, numerous Transformer-based approaches have been introduced for music generation [19], [20], [50], [51], [52], each utilizing unique encoding strategies to capture the complexities of musical information. MusicBERT [52], for instance, employs the OctupleMIDI encoding, which allows for a more granular representation of MIDI events, while SymphonyNet [19] leverages the Multi-track, Multi-instrument Repeatable (MMR) encoding to efficiently represent orchestral music's complexity and diversity.

In recent advancements within the field of music generation, several foundation models (e.g., [20], [53]) have emerged as significant contributors. MusicLM [20] represents a pioneering leap as the first large foundation model capable of generating high-fidelity music based on textual descriptions. This model showcases the ability to understand and translate complex textual prompts into musical compositions, utilizing a hierarchical sequence-to-sequence framework. MusicLM utilizes three key models, including SoundStream [54] for acoustic tokens and w2v-BERT [55] for semantic tokens, along with MuLan [32] for conditioning during training and inference. It has two stages: a semantic modeling stage that learns to map MuLan audio tokens to corresponding semantic tokens, and an acoustic modeling stage that predicts acoustic tokens based on the previously generated tokens and the input text. On the other hand, MusicGen [53], employs an auto-regressive Transformer model that operates over an EnCodec tokenizer [56], with codebooks sampled at 50 Hz, offering a streamlined approach to music generation.

In this paper, we demonstrate our framework's generalizability using MuseGAN, SymphonyNet, and MusicLM, each with distinct structures detailed in Table 2.

3. Threat Model & UEs for Gen. AI

3.1. Threat Model

In the context of generative AI in the music domain, the potential threats are the risks of copyright infringements posed to musicians and the authenticity of their work. To alleviate these concerns, musicians or music composers (i.e., defenders) can employ HARMONYCLOAK to protect their music against unauthorized exploitation by generative AI models, as shown in Figure 2.

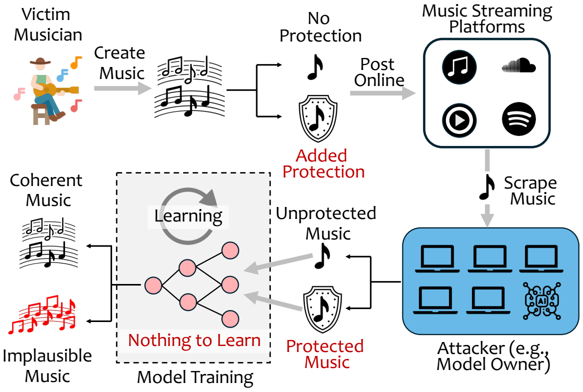


Figure 2: Illustration of the threat model where the attacker scrapes music posted online by victim musicians to train their music generative models.

Attacker's Capabilities. The attacker (e.g. AI companies or model owners) might scrape music data from the Internet or music streaming platforms to train their music-generative AI models, potentially leading to copyright infringements and harming musicians. We assume the attacker possesses substantial advantages and capabilities, including unrestricted access to the training dataset and model parameters, facilitating comprehensive data and gradient inspections, and the ability to perform adaptive attack strategies (e.g., filter or data-augmentation-based), aiming to learn from unlearnable music that is being protected, in response to any perceived degradation in the generative model's performance.

Defender's Objectives. The primary goal of the defender (e.g., music composer) is to render their composed music unlearnable, presenting a significant challenge to generative models attempting to replicate the intricate patterns and structures within the music faithfully. Concurrently, they aim to ensure that these UEs closely resemble the original music and remain indistinguishable for human listeners.

Defender's Capabilities. We assume that the defender has access to a local computing resource (e.g., a laptop) so that he or she can apply defensive imperceptible noises to their music locally before its online distribution. The defender may convert their unlearnable compositions into lossy compression formats (e.g., MP3) for distribution. We also assume that the defender has full access to the portion of composed music that they want to make unlearnable. However, they cannot access the complete training music dataset, limiting their abilities solely to their portion of the data. In this paper, we consider both white-box and black-box settings. Notably, in a **white-box setting**, the defender has comprehensive knowledge of the generative model used to train on the music data, enabling them to leverage it in generating unlearnable music examples. However, the defender cannot interfere with the actual training process itself. In a **black-box setting**, the defender operates with no knowledge of the generative model used for training or the ability to use the generative model to generate music.

3.2. UEs for Generative AI

Unlike creating unlearnable examples (UEs) for classification models, as discussed in Section 2.1, generative models, such as Transformers and GANs, function differently. This distinction necessitates tailored approaches to produce UEs for each type of generative model, due to their unique mechanisms.

Unlearnable Examples for Autoregressive Models. In autoregressive models like Transformers, the objective during training is to minimize the negative log-likelihood (NLL) of the target sequence given the predicted (or partially generated) sequence. Autoregressive models are the core of many recent music generation models, such as MusicLM [20] and SymphoyNet [19]. To reduce the amount of information that can be extracted by the models, we minimize the model's loss by introducing perturbation δ to mislead the model into anticipating a flawless sequence continuation. For a sequence of variables $X_{1:N}$ we achieve this by optimizing the following *min-min* objective function:

$$\min_{\theta} \min_{\delta} -\log \prod_{t=1}^{T} f(x_t | x_{t-1} + \delta_{t-1}, ..., x_{t-p} + \delta_{t-p} : \theta), \tag{3}$$

where x_t represents the predicted value in the sequence at a time t. $x_{t-1}, ...x_{t-p}$ are the previous values in the sequence and p is the autoregressive order. After optimizing for the optimal noise, when we utilize the unlearnable data (i.e., $x+\delta$) to train the autoregressive model, we observe that the model is not generalizing on those samples due to added noise losing the learning capability.

Unlearnable Examples for GAN-based Generative Models. Generative models, such as generative adversarial networks (GANs) [57] (i.e., the core of MuseGAN [18]) are fundamentally different from autoregressive models. They aim to capture the statistical properties of the training data and generate new samples that resemble the training distribution. A GAN has two main components: the generator network (G) and the discriminator network (D). The generator aims to minimize the loss by creating more realistic data, while the discriminator aims to maximize the loss by becoming better at distinguishing real from fake data.

To prevent the generator from extracting the inherent features from a specific training sample x, we can minimize the generator loss by adding an imperceptible noise δ onto the sample to discourage the discriminator from accurately classifying the sample, halting the generator's learning process. This can be achieved through:

$$\min_{G} \max_{D} \left[(\min_{\delta} \mathbb{E}_{(x+\delta)}[\log D(x+\delta)] + \mathbb{E}_z[\log(1 - D(G(z)))] \right], \tag{4}$$

where x represents a real data sample, and z represents the random noise input to the generator. After determining the optimal noise, when utilizing the unlearnable data (i.e., $x + \delta$) to train the GAN, we would observe that the GAN loss reaches a minimum value during the initial training rounds, effectively suppressing the informative knowledge of the data to be learned by the generator.

4. HarmonyCloak

4.1. Methodologies

Design Rationale. As we discussed in Sections 2.3 and 2.4, the defensive noises applied by HarmonyCloak must be meticulously crafted to minimize perceptual quality impact while maintaining effectiveness throughout music production and distribution. Specifically, we focus on the following key aspects to tailor the objective functions in Section 3.2 for creating unlearnable music examples:

❶ *Concealing Noises within Music via Frequency Masking:* To minimize the perceptual impact of the defensive noise on the music, the noise should be subtle and within the *critical bandwidth* of the masking music tunes, which requires its frequencies to be closely aligned with the musical notes.

❷ *Dynamic Variations and Temporal Masking:* Considering the dynamic changes in tempo and volume within the music, the defensive noise must adapt over time and be positioned close to the masking music tunes for effective *temporal masking*, particularly when they cannot be played together.

❸ *Track-Specific Noise Tailoring:* It is essential to create defensive noises tailored to each instrumental track, taking into account their distinct tempo, frequency range, and timbre, for effective concealment within the complex structure of the music.

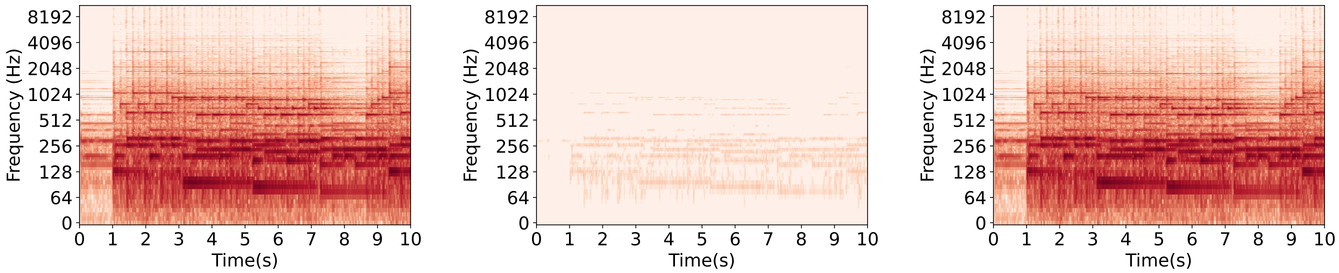
❹ *Remaining Effectiveness under Lossy Compression:* To ensure the effectiveness of defensive noise against lossy compression, it is crucial that the noise not only surpasses the absolute hearing threshold but also aligns with the dominant musical notes (higher masking sound pressure levels, thus achieving a lower MNR), which helps in retaining more noise despite compression.

Constrained Optimization Problem. To generate the imperceptible defensive noise that satisfies all aforementioned requirements, we can employ constrained optimization to solve for the optimal noise. Specifically, given one bar[2] of a multi-track training music sample $\boldsymbol{x} = [x^1, x^2, \ldots, x^M]$ (M tracks), we aim to perturb imperceptible defensive noises crafted for each individual track $\boldsymbol{\delta} = [\delta^1, \delta^2, \ldots, \delta^M]$ onto the sample to make it an unlearnable example, i.e., $\boldsymbol{x} + \boldsymbol{\delta}$, for the generative model. This can be achieved by solving the following bi-level optimization problem:

$$\min_{\theta} \mathbb{E}\Big[\min_{\delta} \mathcal{L}_{gen}\big(f(\boldsymbol{x} + \boldsymbol{\delta})\big)\Big] + \alpha \sum_{m=1}^{M} w^m \|\delta^m\|_2, \tag{5}$$

$$\text{s.t.} \quad \mathcal{H}(T_H) \leq \delta^m \leq x^m, \quad \forall m \in \{1, 2, \ldots, M\} \tag{5a}$$

[2] Note that bars are the basic compositional unit in most music generative models [18], [19], which typically generate music one bar after another. Thus, we consider a bar of the music audio as the basic unit to describe how to generate unlearnable examples.



(a) Clean Music

(b) Defensive Noise

(c) Unlearnable Music

Figure 3: Illustration of the spectrogram of the music and the added defensive noise.

where $f(\cdot)$ denotes the generative model used to train on the music data, $\mathcal{L}_{gen}(\cdot)$ is the generative loss of the model[3], T_H is the absolute human hearing threshold (Equation 2), α is a scaling coefficient, and w^m is defined as one minus the ratio of the cumulative note velocity for track m to the total cumulative note velocity across all tracks. We use w^m to balance the noise intensity added to each instrumental track. $\mathcal{H}(\cdot)$ denotes the function that can be used to convert the threshold in dB SPL to the note velocity as per the guidelines provided in the DLS LEVEL 1 standard [58]:

$$\mathcal{H}(T_H(\nu)) = 127 \cdot 10^{\left(\frac{1}{4}\cdot\log_{10}(T_H(\nu))\right)-L_U+94}, \qquad (6)$$

where L_U is the magnitude of the linear transfer function normalized at 1kHz.

As shown in Equation 5, the inner minimization aims to find the noise that minimizes the overall generative loss on unlearnable music, while the outer minimization problem seeks the model parameters θ (the generator, if the model is GAN-based) that minimize the generative model loss. The objective function's regularizer term aims to minimize the overall amplitude of the added noise by reducing the L2 norm of the noise (for ❶). Additionally, the constraint (Equation 5a) ensures that the noise remains within the masked region by constraining the level between the music (for ❶) and the absolute hearing threshold (for ❹). The bi-level optimization problem is tackled using appropriate algorithms suited for each task. Specifically, we use Projected Gradient Descent (PGD) [59], a first-order optimization method, to solve the constrained inner minimization problem. For the outer minimization, we apply Stochastic Gradient Descent (SGD) [60] with momentum.

Window-based Strategy to Tackle Temporal Dynamics. To align the noise closely with the dominant musical notes (for ❹) while also adhering to the temporal requirement (❷), we divide the music bar ($\boldsymbol{x}$) into short non-overlapped windows during optimization, each window l_w in length. Within each window ($\boldsymbol{x}_t$, as the tth window), we identify the frequency ν_t^m of the dominant musical note for every track (track m), based on their cumulative musical note velocities, and set the defensive noise δ_t^m for the window t and track m to the same frequency ν_t^m as that of the dominant musical note. It is important to note that if the cumulative musical note velocities fall below a certain threshold, indicating the absence of a dominant musical note, then no noise will be introduced. Since the noise is introduced uniquely for each instrumental track, this strategy also facilitates the creation of defensive noises customized for each track (for ❸).

To ensure that the constraint (Equation 5a) can be effectively applied in these windows, we employ a `sigmoid()`-based function to keep the noise δ_t^m within the following range for all windows t and tracks m:

$$\mathcal{H}(T_H(v_t^m)) \le \delta_t^m \le \mathcal{M}(x_t^m), \quad \forall t, m, \qquad (7)$$

where ν_t^m is the dominant frequency, and $\mathcal{M}(\cdot)$ is the velocity of the dominant musical note in the music x_t^m.

Through this constrained optimization and window-based strategy, we can successfully generate imperceptible defensive noise that satisfies all four aforementioned design criteria. Figure 3(a)(c) provides a spectrogram comparison, demonstrating that the clean music and its corresponding unlearnable version are remarkably similar and virtually indistinguishable. Further evidence from Figure 3(b) shows that the defensive noise not only closely aligns with the dominant musical notes but also possesses a much lower intensity compared to the original music, reinforcing our methodology's effectiveness in preserving the original auditory experience.

4.2. HarmonyCloak in Black-Box Setting

In the black-box setting, the defender has neither knowledge about the specific generative model used to train on the music data nor access to use the model to generate music. To make our *error-minimizing* noise applicable to arbitrary generative models regardless of their model type and architecture, we need to improve its cross-model transferability.

Meta-learning [61] is a strategy for tackling new tasks by *learning to learn*. In this approach, a model first learns knowledge and finds connections among multiple training tasks (meta-training phase) and then adapts to the unseen task with a few examples through fine-tuning (meta-testing phase). Inspired by the meta-learning technique, we propose to use a two-step iterative method, as shown in Figure 4, to generate defensive noise to protect music from being learned by unseen and unqueriable generative models. Specifically, we randomly sample $S_1, ..., S_Q$ from a bag of randomly

[3] $\mathcal{L}_{gen}(\cdot)$ serves as a generic loss function and requires updates to align with the specific type of the generative model (Section 3.2).

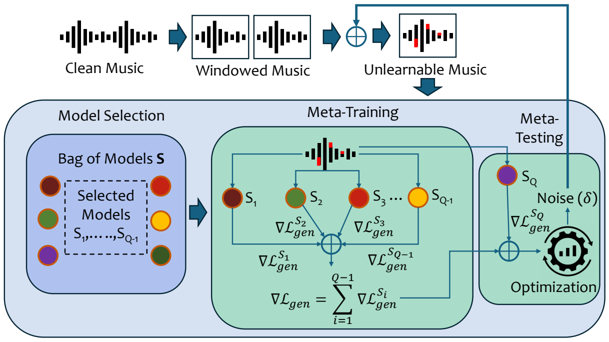

Clean Music
Windowed Music
Unlearnable Music
Model Selection
Meta-Training
Meta-Testing
Bag of Models S
Selected Models
S1,.....,SQ-1
S1
S2
S3 ...
SQ-1
SQ
Noise (δ)
$\nabla\mathcal{L}_{gen}^{S_1}$
$\nabla\mathcal{L}_{gen}^{S_2}$
$\nabla\mathcal{L}_{gen}^{S_3}$
$\nabla\mathcal{L}_{gen}^{S_{Q-1}}$
$\nabla\mathcal{L}_{gen}^{S_Q}$
$\nabla\mathcal{L}_{gen} = \sum_{i=1}^{Q-1}\nabla\mathcal{L}_{gen}^{S_i}$
Optimization

Figure 4: HARMONYCLOAK in black-box settings.

initialized surrogate models $\boldsymbol{S}$ and perform the following meta-training and meta-testing to compute defensive noise.

Meta-training. After segmenting the music into windows and initializing the defensive noise δ_t^m for the window t and track m, we utilize the first Q-1 models from the bag to simulate a white-box scenario, thereby generating unlearnable music. Consequently, the bi-level optimization, represented by Equation 5, transforms into the following formulation:

$$\min_{\theta}\ \mathbb{E}\left[\min_{\delta}\left(\sum_{q=1}^{Q-1}\mathcal{L}_{gen}^{q}(f(x+\delta\})\right)\right]+\alpha\sum_{m=1}^{M}w^m\|\delta^m\|_2 \quad (8)$$

$$\text{s.t.}\quad \mathcal{H}(T_H)\leq\delta^m\leq x^m,\quad \forall m\in\{1,2,\dots,M\} \quad (8a)$$

where q is the surrogate model in the bag ($q\in\boldsymbol{S}$). As we have multiple gradients from the selected models, we take the average of the gradients by dividing it by the total number of models to ensure balanced optimization and stability, accurately scaling each model's contribution across both meta-training and meta-testing phases:

$$\nabla_x\mathcal{L}_{gen}(f(\boldsymbol{x}+\boldsymbol{\delta}))=\frac{1}{Q}\sum_{q=1}^{Q-1}\mathcal{L}_{gen}^{q}(f(\boldsymbol{x}+\boldsymbol{\delta})). \quad (9)$$

Meta-testing. After the meta-training phase, we transition to meta-testing to refine the noise for adaptation to the unseen model. During this stage, we fine-tune the defensive noise for the last sampled model S_Q in a black-box setting to bolster its generalizability. In this black-box setting, where direct access to the model's loss function is unavailable, we utilize divergence loss as a surrogate metric. Divergence loss quantifies the disparity between the distribution of the target model's output and the distribution of the clean music, providing a measure of how effectively the perturbation minimizes the learning. Following this, the process for crafting defensive noise aligns with the optimization objectives outlined in Section 4.1. This involves iteratively adjusting the perturbation to minimize the model's loss while maintaining imperceptibility to human perception. Through this iterative optimization process, guided by the divergence loss in the black-box setting, we refine the defensive noise to maximize its effectiveness across a variety of unseen models, thus enhancing its generalizability and robustness.

4.3. HARMONYCLOAK for Wave Audio

Our methodologies introduced in Sections 4.1 and 4.2 primarily employ the MIDI audio format to create unlearnable examples. However, many recent music-generative AI models (e.g., MusicLM [20] and MusicGen [53]) operate on wave-based audio formats, such as *.wav*, where different instrumental tracks are amalgamated to produce music. Our approach is adeptly adaptable to these wave-based formats without altering the core objective function, which remains consistent as outlined in Equations 5 and 8, respective to the defense scenario at hand.

In adapting our method for wave-based formats, we use the same window-based strategy to divide music into smaller windows, each with 10ms length, and then apply the Short Time Fourier Transform (STFT) to ascertain the dominant frequencies (ν_t) within each window (window t). The absolute hearing threshold at these dominant frequencies ($T_H(\nu_t)$), presented in dB SPL, can be calculated by Equation 2. We can then compute the magnitude, $M(\nu_t)$, of the music at each dominant frequency using the Fast Fourier Transform and calculate the relative dB SPL [44] using:

$$\mathbb{M}_t=-20\cdot log\left(\frac{M(\nu_t)}{20^{-6}}\right). \quad (10)$$

To ensure the defensive noise aligns closely with the dominant musical notes, noise δ_t is introduced at the dominant frequency within each window. The magnitude of this noise is maintained within the following range to meet our aforementioned design criteria for all windows t:

$$T_H(\nu_t)\leq\delta_t\leq\mathbb{M}_t,\quad \forall t. \quad (11)$$

It is important to note that when using multi-track wave-based audio for training generative models (e.g., MusicLM [20] supports both single-track and multi-track wave-based training music), the aforementioned operations are applied to each track individually. The subsequent process for creating defensive noise follows the optimization objectives introduced in Sections 4.1 and 4.2. This modification tackles the variety of formats for presenting musical notes, highlighting the adaptability of HARMONYCLOAK across both MIDI and wave-based audio formats without compromising on the process' integrity.

5. Evaluation

5.1. Experimental Setup

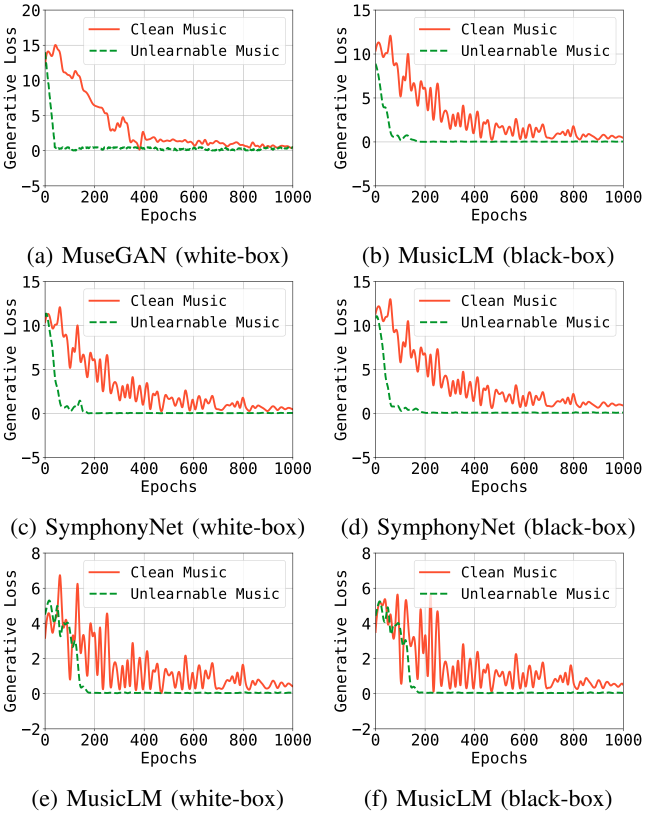
Evaluation Metrics. To assess the performance of HARMONYCLOAK, we analyze it from two key perspectives: *(1) Effectiveness*: Effectiveness is measured by both training loss and the perceptual quality of the music produced by generative models when trained using unlearnable examples, showing the impact of HARMONYCLOAK on the model's ability to generate high-quality music; and *(2) Perceptual Quality*: The generated music UEs should have an enjoyable listening experience for the audience, and the introduced defensive noises should remain a minimal impact on the music perceptual quality.

To evaluate the quality of both the unlearnable music examples and the AI-generated music, we adopt the following intra-track and inter-track metrics that are commonly used in the field of music by prior studies [18], [20]:

- **Empty Bars (EB) Ratio:** It calculates the percentage of empty bars in the generated music, meaning the percentage of the silent part of the music compared to the whole music.
- **Used Pitch Classes (UPC) Per Bar:** It measures the number of unique pitch classes utilized within each bar of the generated music. It ranges from 0 to 12, representing the 12 pitch classes in music. The UPC metric helps assess the diversity and variety of pitches employed in music.
- **Qualified Notes (QN) Ratio:** It quantifies the percentage of qualified notes in the generated music. A qualified note refers to a note with a duration of at least three-time steps. The QN ratio provides insights into the level of fragmentation or coherence in the music.
- **Drum Pattern (DP):** It focuses on drum tracks and measures the ratio of notes that conform to standard 8- or 16-beat patterns. The DP metric indicates the adherence to established rhythmic patterns in the generated drum tracks.
- **Tonal Distance (TD):** TD [62] quantifies the tonal distance between a pair of tracks. A larger TD value implies weaker inter-track harmonic relations. This metric helps assess the level of harmonicity or dissonance between different tracks in the generated music. It ranges from $0-5$, where the lower bound of 0 represents maximum inter-track harmonic relations and the upper bound of 5 represents weaker inter-track harmonic relations or maximum tonal dissimilarity.
- **Fréchet Audio Distance (FAD):** FAD [63] compares statistics computed on a set of reconstructed music clips to background statistics computed on a large set of studio-recorded music. This is a reference-free audio quality metric, which correlates well with human perception. Music with a low FAD score is more plausible to the human ear.

TD and FAD are used to measure perceptual quality as they focus on the overall quality and perceptual fidelity by measuring the harmonic relations and statistical characteristics. EB, UPC, QN, and DP are used to compare the real music samples with the generated samples in the temporal domain, providing insights into the generators' performance (i.e., effectiveness). When the distributions of real music samples and generated samples are similar, it follows that the temporal domain metrics should also exhibit proximity.

Experimental Settings. In this work, we evaluate HarmonyCloak on three state-of-the-art music generative models, i.e., MuseGAN [18], SymphonyNet [19], and MusicLM [20]. We use Lakh MIDI Dataset [64], which contains 45,129 MIDI files featuring music from up to seven different instruments and spanning a wide range of genres, to train the generative models and generate unlearnable examples. By default, unless specified otherwise, we deliberately introduce noise to only 15% of the complete training dataset to demonstrate the practicality of our approach. Unless mentioned otherwise, we set the window size l_w to 10ms for all the models. The 10ms window length balances temporal resolution with stability, as significant tune changes are rare in such a short period. Minor variations are managed by focusing on the dominant frequency, keeping noise aligned with key musical features. How the window size affects the performance is discussed in Section 5.2.5.



(a) MuseGAN (white-box) (b) MusicLM (black-box)

(c) SymphonyNet (white-box) (d) SymphonyNet (black-box)

(e) MusicLM (white-box) (f) MusicLM (black-box)

Figure 5: The training generative loss curves of HarmonyCloak.

For the optimization tasks, we use a step size of 0.001 for PGD, running it for 20 iterations, while for SGD, we choose a momentum of 0.9 and an initial learning rate of 0.025. For MusicLM, as it works on waveform-based music we convert the Lakh MIDI dataset to WAV format (16kHz, 16-bit PCM, Mono) for training. In each setting, we generate 5,000 bars of music using each model for evaluation. For MuseGAN and MusicLM, we use Adam [65] optimizer with a learning rate of 10^{-4}, for SymphonyNet, we use AdamW [65] optimizer with a learning rate of 3×10^{-4}. For MuseGAN, we use 0.22 for α, and for SymphonyNet and MusicLM, we use 0.30. In the black-box setting, we use a total of 10 randomly initialized MuseGAN models, featuring diverse architectures and varying numbers of parameters for the generator and discriminator. In each round of training, we randomly select 7 models from the bag for the defensive noise calculation.

TABLE 3: Intra-track performance evaluation of HarmonyCloak against generative models.

Training Music	Model	Empty Bars (EB;%)					Used Pitch Classes (UPC)				Qualified Notes (QN;%)				Drum Pattern
		B	D	G	P	S	B	G	P	S	B	G	P	S	(DP; %)
Training Data	-	8.95	8.19	20.2	23.2	11.5	1.92	4.78	4.37	4.77	87.2	83.2	82.5	87.6	89.6
Clean Music	MuseGAN	6.59	2.33	18.3	22.6	6.10	1.53	3.69	4.13	4.09	71.5	56.6	62.2	63.1	93.2
	SymphonyNet	7.29	8.32	19.7	22.2	10.6	1.59	4.75	4.13	4.22	81.9	72.0	71.1	81.5	88.0
	MusicLM	8.22	7.93	20.5	22.5	11.1	1.89	4.66	4.84	4.39	86.4	80.5	80.6	85.2	88.1
Unlearnable Music (white-box)	MuseGAN	0.01	0.05	0.02	0.02	0.02	11.8	11.3	11.1	11.3	64.2	63.7	67.3	55.2	61.9
	SymphonyNet	0.01	0.01	0.01	0.02	0.07	10.9	9.2	11.6	9.2	51.2	55.6	54.7	58.2	66.1
	MusicLM	4.39	4.22	5.5	7.2	2.19	7.6	7.5	6.4	8.9	66.9	71.6	69.2	67.4	74.4
Unlearnable Music (black-box)	MuseGAN	0.01	2.75	1.09	0.02	0.02	10.9	10.2	9.7	10.7	62.2	63.4	64.2	60.3	77.2
	SymphonyNet	0.05	2.21	1.23	0.09	0.16	10.2	10.1	10.7	10.5	55.1	92.3	93.3	97.2	72.3
	MusicLM	1.22	4.19	2.45	1.10	0.89	8.22	8.12	8.67	8.81	67.2	66.4	64.2	61.4	77.1

TABLE 4: Harmonicity comparison of generated music (**B:** Bass, **G:** Guitar, **P:** Piano, **S:** Strings).

Training Music	Model	Tonal Distance (TD)					
		B-G	B-S	B-P	G-S	G-P	S-P
Clean Music	MuseGAN	1.66	1.71	1.63	1.42	1.44	1.35
	SymphonyNet	1.06	1.13	1.26	1.22	1.12	1.03
	MusicLM	1.27	1.38	1.34	1.15	1.32	1.41
Unlearnable Music (white-box)	MuseGAN	3.53	2.94	3.02	3.10	3.02	4.04
	SymphonyNet	2.82	3.72	4.94	4.42	4.39	4.13
	MusicLM	2.17	2.28	2.53	2.75	2.62	2.61
Unlearnable Music (black-box)	MuseGAN	3.62	3.24	3.53	3.19	3.23	4.14
	SymphonyNet	3.12	3.93	4.72	4.18	4.22	4.23
	MusicLM	3.01	2.95	3.62	3.18	3.11	3.93

TABLE 5: Performance of HarmonyCloak in preserving music quality.

Music	Tonal Distance (TD)						Avg. FAD Score
	B-G	B-S	B-P	G-S	G-P	S-P	
Clean Music	1.57	1.58	1.51	1.10	1.02	1.04	1.23
Unlearnable Music (white-box)	1.65	1.66	1.69	1.11	1.09	1.09	1.31
Unlearnable Music (black-box)	1.69	1.69	1.71	1.13	1.10	1.11	1.35

5.2. Results

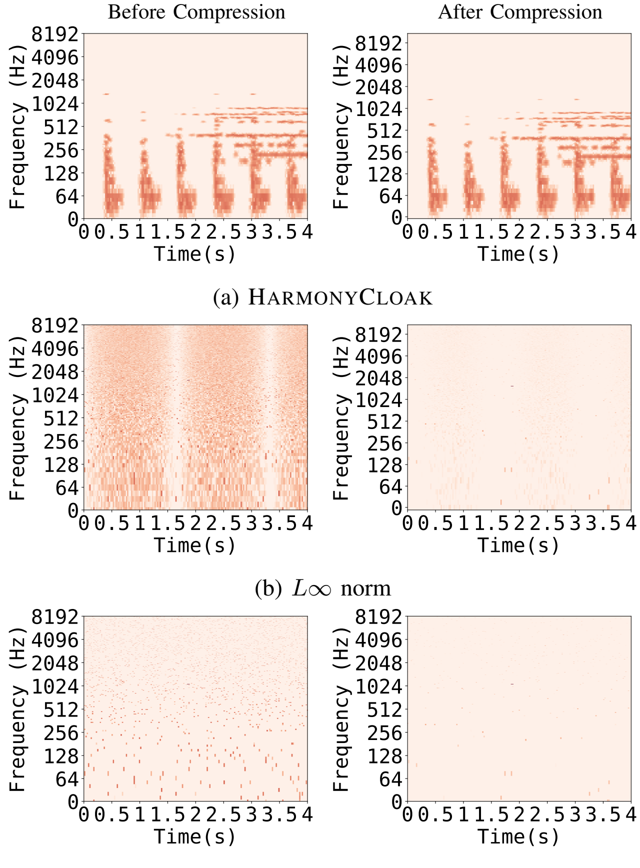
5.2.1. Effectiveness. Effectiveness is evaluated through the following aspects:

Training Loss Comparision. To demonstrate the effectiveness of the generated defensive noise, we examine the training loss curves for both clean and unlearnable music examples in white-box and black-box settings. Unlearnable examples can cause the model's generative loss to approach zero, misleading the model into thinking there is nothing further to learn. As shown in Figure 5, we find that, for three target models, the training loss on unlearnable music examples quickly approaches zero after a few iterations in both white-box and black-box settings. When compared to MuseGAN and SymphonyNet, unlearnable examples on MusicLM show a relatively higher loss. This discrepancy is attributed to MusicLM's two-step audio processing which makes it harder to minimize the loss with constrained noise. Overall, HarmonyCloak's effectiveness is evident in both white-box and black-box settings, as observed by the training loss on unlearnable music examples maintaining a value close to zero across both scenarios.

Temporal Analysis. In our temporal analysis of the generated music, as presented in Table 3, we conducted a comprehensive analysis of various aspects of the generated music to assess its quality and performance. Notably, models trained on unlearnable music exhibit a significantly lower percentage of EB compared to models trained on clean music and even the training data, indicating a lack of rhythm and structure in the generated music. Moreover, higher UPC values in the models trained with unlearnable music indicate the utilization of nearly all pitch classes in each instrument, making the music sound implausible to human listeners. Additionally, the percentage of QN differs between models trained on unlearnable music and clean music, implying that the generated notes are either shorter or longer compared to clean musical notes, further contributing to the perceived lack of musicality in generated music. Lastly, in terms of DP, the generated music performs poorly, implying that the model struggles to generate convincing and coherent drum patterns. This deficiency in generating realistic drum patterns can significantly impact the overall quality and authenticity of the music. From the results, it's evident that MusicLM slightly outperforms MuseGAN and SymphonyNet. Despite aiming to prevent unauthorized exploitation of music data, our results suggest that even a small portion of unlearnable examples in the training data can hinder learning.

Perceptual Analysis. Table 4 presents a comparison of the TD distances between the models when trained with clean and unlearnable music, respectively for both scenarios. The table shows that all three models exhibit lower TD values when trained with clean music, implying a more substantial inter-track harmonic relation, which indicates better-sounding music. However, all the models show significantly higher TD values when trained with unlearnable music irrespective of the training scenario, implying weaker inter-track harmonic relation signifying less harmonic music.

5.2.2. Perceptual Quality. In Table 5, we compare the Tonal Distance (TD) between the clean and unlearnable music in both white-box and black-box settings. We observe a slight increase in TD value among different tracks, indicating weaker harmonic relationships. However, these minor



(a) HarmonyCloak

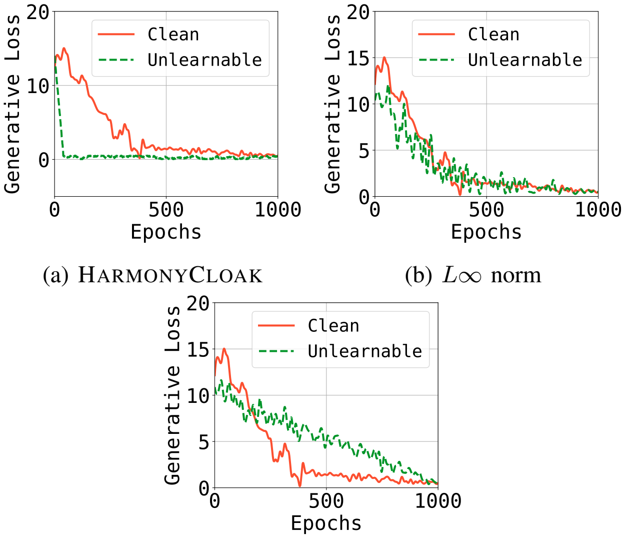
(b) $L\infty$ norm

(c) $L2$ norm

Figure 6: Comparison of the generated defensive noise before and after MP3 compression.

changes are unlikely to significantly impact the overall quality of the original music [62]. One key observation is that the unlearnable music generated in the black-box setting has a higher TD than the white-box, attributed to the lack of direct knowledge about the target model. Considering that the original music is intended for a wide audience, we evaluate its quality using metrics that closely align with human perception. We report the FAD score based on the VGGish [66] audio embedding model, which has been trained on the YouTube-8M audio event dataset [67]. The FAD score indicates that the unlearnable and clean music exhibit similar plausibility, affirming the introduced noise's limited perceptual impact.

5.2.3. Resilience against Lossy Compression. To demonstrate that the defensive noises introduced by HarmonyCloak are robust against lossy compression process, we compare the defensive noise generated by HarmonyCloak with the noises produced under $L\infty$-norm and $L2$-norm constraints on the target model MuseGAN rather than our psychoacoustic modeling based approach, before and after MP3 compression. As shown in Figure 6, we find that the noises generated under L_p-norm constraints overwhelm the original music's pitches and harmonics, masking them almost entirely. However, when MP3 compression is applied, this noise is mostly eliminated. The MP3's low-resolution encoding for masked noise effectively eliminates the noise, leaving the original music relatively unscratched. From the training loss in Figure 7, we can also observe that, after compression, the model trained on this L_p-norm based unlearnable examples exhibits significantly higher training loss, implying the model can effectively learn from these samples. Differently, HarmonyCloak generates noise that harmonizes with the music frequencies and falls within the masked region, making them less perceptible. Our technique aims to minimize the Masking-to-Noise Ratio (MNR) to ensure that, even after compression, the noise maintains its core characteristics, preserving the samples' unlearnability.



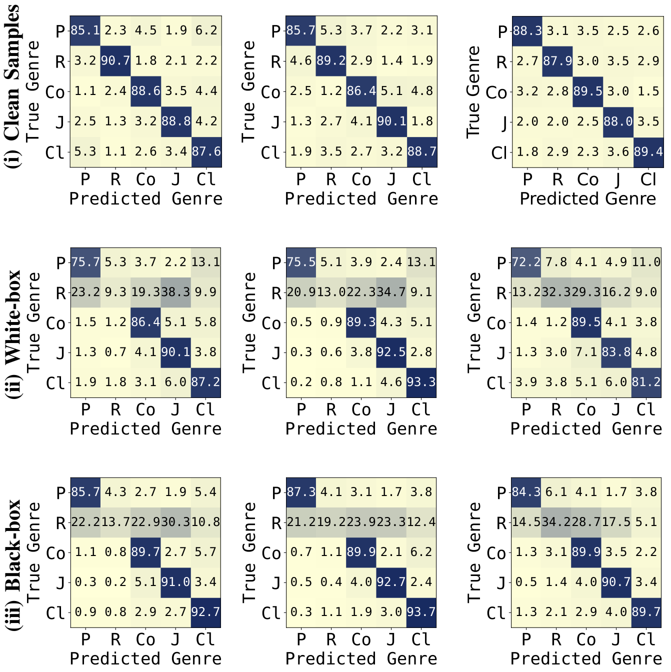
(a) HarmonyCloak

(b) $L\infty$ norm

(c) $L2$ norm

Figure 7: Comparison of unlearnability of generated defensive noise before and after MP3 compression.

5.2.4. Unlearnability Analysis on Genre Classification. We further conduct genre unlearnability evaluation, in which we generate unlearnable examples only for the Rock (R) music in our dataset, and keep other genres like Pop (P), Classical (Cl), Country (Co), and Jazz (J) intact. Subsequently, we train a music genre classifier using the clean dataset and employ it to predict the genre of music generated by our generative models. This evaluation helps us assess whether the models can replicate patterns of perturbed rock music. For MuseGAN, we adopt a track-conditional setting, similar to the method outlined by [18]. Specifically, we provide piano tracks as conditions and generate the remaining four tracks. During the evaluation phase, we feed piano tracks from various genres of music into the model and assess the generated music using the genre classifier. SymphonyNet, designed to complete partial music pieces, takes a different approach. We provide SymphonyNet with a 300ms snippet of music from different genres and obtain the complete music pieces for evaluation. For MusicLM, we provide a brief description of the music genre to generate



(a) MuseGAN (b) SymphonyNet (c) MusicLM

Figure 8: Genre unlearnability analysis.

music. In Figure 8, the confusion matrix of the genre classifier reveals that the classifier struggles to confidently label rock music as “Rock” in both white-box and black-box settings. Compared to the classification performance of the model trained with clean samples, we also observe that the presence of unlearnable examples in the training set has a limited impact on its performance across other genres. Additionally, SymphonyNet demonstrates slightly higher confidence compared to MuseGAN, mainly because some of the generated music aligns with the given genre conditions. MusicLM exhibits the highest confidence among all three because it includes pre-trained components that are not part of the unlearnable music generation process. These results indicate that the generative models have only acquired limited knowledge from the perturbed rock genre in the unlearnable examples, especially when trained on the unlearnable music generated using our method. However, this does not hinder the model’s ability to learn other genres, as it may still generate common features of the protected genre that overlap with those of other genres.

5.2.5. Impact of Window Size. In the design of HARMONYCLOAK, we use a window-based strategy to address temporal dynamics. Determining the optimal window size is crucial, as it ensures sufficient noise is added to the music to render it unlearnable while aligning the noise with the dominant frequency of the music for effective masking. To identify the appropriate window size, we conducted an ablation study on MuseGAN in the white-box setting with varying window sizes. Figure 9 presents the unlearnability analysis of the generated music for different window sizes, showing the loss curves for each. The figure clearly indicates that shorter window sizes result in steeper loss curves, which more effectively compress the knowledge that can be learned during training. Table 6 supports this finding, showing that shorter window sizes lead to much higher FAD scores of the generated music, indicating low music quality. However, the impact of window size on the FAD scores of unlearnable music is minimal. Therefore, even with increased noise levels associated with shorter window sizes, the quality of unlearnable music remains relatively consistent with that of clean music. Consequently, we chose a 10ms window size, as it achieves a balance by maintaining a noise level similar to smaller windows while ensuring the FAD score remains high enough to prevent the generative model from effectively learning from the music.

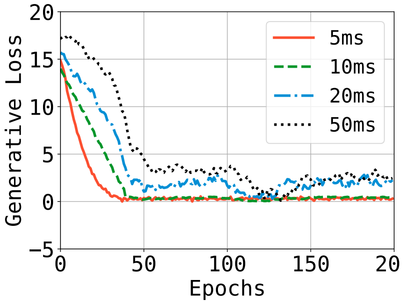


Figure 9: Impact of different window sizes on the unlearnability.

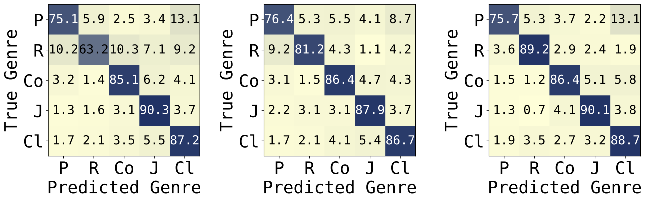
TABLE 6: Impact of different window sizes on the perceptual quality and running time.

Window Length	Avg. FAD (Generated Music)	Avg. FAD (Unlearnable Music)
5ms	11.3	1.3105
10ms	10.9	1.3179
20ms	8.7	1.2818
50ms	6.8	1.2798
100ms	4.2	1.2522

5.2.6. Impact of Unlearnable Percentages. Unlearnable music effectively tricks the generative model into perceiving those samples as devoid of useful information. However, this raises a question: do these unlearnable samples affect the learning process for other clean samples with similar features, such as music from the same genre? To investigate this, we conducted an experiment where varying percentages of rock music were rendered unlearnable, and then we trained the MuseGAN model using a partially unlearnable ($\mathcal{D}_u$) and partially clean ($\mathcal{D}_c$) rock music set combined with clean music set of other genres. Figure 10 presents the confusion matrix for the genre classifier’s inference on the music, which shows that the presence of unlearnable music does not hinder the generative model’s ability to learn from the clean samples. However, the classification accuracy for the “Rock” category greatly declines when 80% of the examples are unlearnable, primarily because the remaining

TABLE 7: Impact of unlearnable percentages on the generated music. Percentage of unlearnable examples is $\frac{\mathcal{D}_u}{\mathcal{D}_u+\mathcal{D}_c}$.

Percentage of Unlearnable Music	Empty Bars (EB;%)					Used Pitch Classes (UPC)				Qualified Notes (QN;%)				Drum Pattern (DP; %)
	B	D	G	P	S	B	G	P	S	B	G	P	S	
Clean Training Music	8.95	8.19	20.2	23.2	11.5	1.92	4.78	4.37	4.77	87.2	83.2	82.5	87.6	89.6
0%	6.59	2.33	18.3	22.6	6.10	1.53	3.69	4.13	4.09	71.5	56.6	62.2	63.1	93.2
5%	5.97	2.44	18.4	21.3	6.01	1.58	3.71	4.33	5.03	70.2	63.7	61.3	59.2	88.9
10%	3.27	1.75	7.89	0.16.3	3.62	4.34	5.23	6.73	8.71	65.1	61.9	64.2	60.1	78.5
15%	0.01	0.05	0.02	0.02	0.02	11.8	11.3	11.1	11.3	64.2	63.7	67.3	55.2	61.9



(a) 80%

(b) 50%

(c) 20%

Figure 10: Impact of unlearnable percentages on genre classification. Percentage of unlearnable examples is $\frac{\mathcal{D}_u}{\mathcal{D}_u+\mathcal{D}_c}$ for rock genre.

20% clean rock music examples do not provide enough data for the model to perform effectively. We observe a similar trend when different percentages of other types of music are rendered unlearnable. For example, making 80% of pop music unlearnable reduces the classifier's accuracy to 57.8% compared to 75.1% when only 20% is unlearnable. Similarly, the accuracy drops to 65.7% for jazz, 59% for country, and 47% for classical music.

Additionally, we conducted a study using MuseGAN with different percentages of randomly chosen unlearnable music in the training set ($\mathcal{D}_u+\mathcal{D}_c$) to test how it hampers the overall learning of the generative model. Table 7 shows that as the proportion of unlearnable examples in the training data increases, there is a corresponding decrease in the model's performance compared to the clean training data, as measured by EB, UPC, QN, and DP, on the generated music. In contrast to classification models, which primarily focus on distinguishing between categories, generative models strive to capture the underlying data distribution. They learn patterns, structures, and relationships within the training data to produce new, similar outputs. Therefore, a reduction in the proportion of learnable music samples compromises the model's capacity to generate plausible music.

6. Subjective User Study

We conducted a listening test[4] with a diverse group of 31 participants (21 males, 10 females) aged between 25 and 36. All participants in the test were self-identified music lovers. Throughout the test, each subject was presented with four sets of music clips in a randomized order to minimize potential bias. Each music clip has a minimum duration of 30 seconds. The first set contains two versions of the training music: the original clean version and its corresponding unlearnable version, which was generated using both white-box and black-box settings. The other sets featured four music samples: two from the target model (MuseGAN, SymphonyNet, and MusicLM) trained on clean music and two from the target model trained on unlearnable music. Each subject was asked to rate the music in terms of whether the music 1) has pleasant harmony (H↑); 2) plausibility (P↑); 3) presence of noise (N↑); and 4) the overall rating (OR↑) on a 5-point Likert scale. Sample audio clips can be found on the anonymized website [68].

TABLE 8: Performance of HarmonyCloak (**H:** Harmonious, **P:** Plausibility, **N:** Noisy, **OR:** Overall Rating) in user study.

Settings	Music	Model	Avg. User Rating			
			H↑	P↑	N↑	OR↑
	Clean Training Music	-	4.17	4.81	4.19	4.45
	Generated Music (Trained on Clean Music)	MuseGAN	4.11	4.05	4.44	4.17
		SymphonyNet	4.44	4.53	4.20	4.11
		MusicLM	4.55	4.49	4.29	4.31
White-box	Unlearnable Training Music	-	4.11	4.72	4.01	4.32
	Generated Music (Trained on Unlearnable Music)	MuseGAN	2.34	3.10	1.20	2.40
		SymphonyNet	2.20	2.94	2.25	2.22
		MusicLM	3.22	3.10	2.29	2.31
Black-box	Unlearnable Training Music	-	4.03	4.45	3.98	4.12
	Generated Music (Trained on Unlearnable Music)	MuseGAN	2.44	3.78	1.94	2.88
		SymphonyNet	2.20	2.94	2.45	2.22
		MusicLM	3.41	3.04	2.78	2.34

Table 8 provides an overview of the ratings obtained from the user study. The results indicate that the clean and the unlearnable training samples generated using white-box settings received similar ratings from the users. For example, the H scores for these samples are 4.17 and 4.11, respectively, while the OR scores are 4.45 and 4.32, respectively. Additionally, unlearnable music samples generated in black-box settings received slightly lower ratings, indicating a minor drop in music quality due to a lack of knowledge about the target generative model. However, their H, P, and OR scores are still above 4. User ratings for models trained on unlearnable music, such as MuseGAN, SymphonyNet, and MusicLM, received significantly lower scores compared to those trained on clean music, with OR scores as low as 2.22, 2.31, and 2.40 in white-box settings. These findings underscore the substantial impact of unlearnable music on the generated music's quality and perception.

[4] The experiments and interviews were formally classified as exempt by our university's IRB.

TABLE 9: Robustness of HarmonyCloak against existing noise removal techniques.

Music	**Defense Method**	**Empty Bars (EB;%)**					**Used Pitch Classes (UPC)**				**Qualified Notes (QN;%)**				**Drum Pattern (DP;%)**
		B	**D**	**G**	**P**	**S**	**B**	**G**	**P**	**S**	**B**	**G**	**P**	**S**	
Clean Music	-	8.95	8.19	20.2	23.2	11.5	1.92	4.78	4.37	4.77	87.2	83.2	82.5	87.6	89.6
Generated Music on Unlearnable Music	-	0.01	0.05	0.02	0.02	0.02	11.8	11.3	11.1	11.3	64.2	63.7	67.3	55.2	61.9
	SS [69]	1.30	4.15	2.05	0.55	0.80	11.0	8.70	11.2	10.4	69.5	68.8	71.5	62.7	74.0
	NMF [70]	0.82	4.05	2.25	0.95	1.08	11.0	10.3	11.1	10.2	68.8	68.7	71.2	61.5	67.1
	EPIc [71]	0.60	3.20	1.58	0.06	0.05	11.1	9.30	10.3	10.4	67.5	67.6	68.4	61.2	63.5
	DP-InstaHide [72]	1.28	4.15	2.05	0.50	0.80	11.0	8.70	11.1	10.3	69.5	68.8	71.5	62.7	74.1

7. Robustness of HarmonyCloak

In HarmonyCloak, incorporating imperceptible noise plays a vital role in making the music unlearnable for generative AI models. To show the robustness of our generated unlearnable music examples against existing noise reduction algorithms and defensive strategies against *data poisoning attacks*, we evaluate HarmonyCloak under two prominent noise reduction techniques, i.e., Spectral Subtraction (SS) [69], and Non-negative Matrix Factorization (NMF) [70] and two state-of-the-art defenses EPIc [71] and DP-InstaHide [72]. SS focuses on estimating and subtracting noise spectrum from the observed signal. NMF is a matrix decomposition technique widely used for audio source separation and noise reduction. By decomposing the observed signal into its constituent parts, NMF can isolate the noise components and reconstruct a cleaner version of the original signal. EPIc detects and eliminates poisoned data while DP-InstaHide augments the data to eliminate poisoning.

Table 9 presents the EB, UPC, QN, and DP values of the music samples generated by the model trained on unlearnable examples and the samples filtered by these noise-removal techniques, respectively. The findings suggest that despite the adoption of these prominent noise removal techniques and adaptive defenses, the generative model still struggles to produce high-quality music. For instance, the inter-track and intra-track values of music samples generated by the model trained on filtered unlearnable examples still significantly differ from clean music. For instance, the Bass track has an EB score of 0.82 when applying the NMF filter, whereas the clean music scores 8.95. While using EPIc, the EB percentage for the Guitar track in the generated music increases from 0.02 to 1.58, and the model trained on clean music achieves a much higher score of 20.2. These disparities strongly suggest that the generated music remains implausible despite undergoing specific filtering techniques or adaptive defense mechanisms. This observation highlights the resilience of HarmonyCloak against various noise removal techniques, as the generated samples consistently lack plausibility.

8. Related Work

Generative AI in Music. The history of machine-generated music dates back to the 1950s with works like the ILLIAC Suite by Hiller [73]. The first AI-based music models with neural networks emerged in the late 1980s by Nierhaus [74]. In recent years with the rise of generative models, music generation become a topic of interest for researchers. Roberts *et al.* proposed MuseVAE [75], which uses a hierarchical decoder for variational autoencoder (VAE) models to address the challenges of modeling the long-term structure in sequential data like musical nodes. Dhariwal *et al.* [76] proposed Jukebox, which utilizes a multiscale VQ-VAE to compress long-context raw audio and employs Autoregressive Transformers to generate high-fidelity songs with music. MuseGAN [18] was the first multi-track generative model, where three Generative Adversarial Networks (GAN) are used to address the unique challenges of generating music, including temporal dynamics, interdependence between tracks, and the ordering of musical notes. More recently, SymphonyNet [19] proposed by Liu *et al.*, is a permutation invariant language model for symphonic music generation, utilizing a modified Byte Pair Encoding algorithm.

Additionally, in the domain of music generation conditioned on text, Mubert [77] employs a Transformer to embed text prompts and select music tags that closely match the encoded prompt to query a song generation API. On the other hand, Riffusion [78] fine-tunes a Stable Diffusion model on mel spectrograms of music pieces from a music-text dataset, offering a distinct approach to generating music based on textual input. Most recently, Agostinelli *et al.* proposed MusicLM [20], a high-fidelity music generation model that leverages autoregressive modeling and text conditioning. They achieve this by projecting music and text descriptions into a shared embedding space, enabling training on audio-only data and conditioning on text during inference.

Unauthorized Data Usage Prevention in AI. There has been active research to prevent unauthorized use of data. Fowl et al. conducted studies showcasing the efficacy of adversarial examples in data poisoning, surpassing previous poisoning methods concerning secure dataset release [29]. The exploration of indiscriminate poisoning attacks, introduced by Yu *et al.* [79], has emerged as a preventive strategy against unauthorized data exploitation. Their research focuses on the examination of the linear separability of sophisticated attack perturbations when associated with target labels. To prevent third-party training on data without permission, Shan *et al.* proposed a privacy protection method that leverages targeted adversarial attacks to add imperceptible perturbations to users' data, rendering models trained on the perturbed dataset invalid and protecting privacy against unauthorized deep learning models [10].

More recently, Huang *et al.* [11] and Fu *et al.* [12] proposed *unlearnable* strategies using error-minimizing noise, reducing the error of training examples to make them unlearnable. Liu *et al.* [80] improved the robustness of unlearnable examples by leveraging data's grayscale knowledge. Furthermore, Ren *et al.* [13] used Classwise Separability Discriminant (CSD) to enhance the transferability of unlearnable examples across different training settings and datasets. Zhao *et al.* [81] proposed unlearnable examples for diffusion models using the error-minimizing noise strategies. Zhang *et al.* [82] proposed label-agnostic unlearnable examples with cluster-wise perturbations. Liu *et al.* introduced stable unlearnable examples by training the defensive noise against random perturbation instead of the adversarial perturbation to improve the stability of defensive noise [83]. Similar to unlearnable examples, Ye *et al.* proposed ungeneralizable examples [84], which are trained by maximizing a designated distance loss in common feature space with the addition of undistillation optimization.

In addition to unlearnable examples, other research has explored various protective techniques. For example, Chen *et al.* have developed EditShield [85], a method that introduces distortions to protect against unauthorized image editing by misleading instruction-guided diffusion models. Similarly, Liu *et al.*'s MetaCloak [86] uses meta-learning to generate robust, transformation-resistant perturbations aimed at protecting personal data from misuse in text-to-image synthesis. Furthermore, Shan *et al.* proposed Glaze [17], a method designed to protect artists from style mimicry by text-to-image diffusion models. This is achieved by introducing perturbations that maximize the feature differences from the original image. Additionally, Shan *et al.* introduced Nightshade [87], a prompt-specific poisoning attack for diffusion-based text-to-image models capable of causing related concept destabilization. Lastly, emphasizing the unique application in speech data, Yu *et al.* proposed AntiFake [88], an adversarial audio system designed to thwart unauthorized speech synthesis, safeguarding the integrity of audio content from exploitative AI technologies.

Although the aforementioned studies make valuable contributions to prevent unauthorized data usage, they primarily concentrate on vulnerabilities in deep learning models related to image classifications or image style mimicry with some efforts in the speech domain. There is a notable gap in research regarding generative AI models, particularly in their application to creating music artworks.

9. Discussion and Future Work

Expanding to Vocal-Instrumental Compositions. Our current implementation of HarmonyCloak primarily addresses instrumental music, which, while complex, does not fully encompass the challenges posed by vocal performances. Vocals bring additional layers of complexity due to intricate harmonic structures, varying timbres, and dynamic temporal characteristics. These nuances may not be fully addressed by the current approach, which focuses on instrumental soundscapes. Future research will extend HarmonyCloak to handle vocal-instrumental compositions, applying imperceptible noise to both types of audio elements. This work will explore how defensive techniques can protect vocals without distorting their tonal qualities, ensuring that artists' expressive voices, both literal and metaphorical, remain safeguarded from AI-driven content generation while retaining the artistic quality of the music.

Involving Professional Musicians for Deeper Insights. Our feedback collection so far has been limited to general music enthusiasts, whose appreciation and understanding of music may lack the depth and critical insights offered by professional musicians. As a result, certain subtleties and practical concerns that are crucial to real-world adoption may have been overlooked. For future studies, we will actively involve professional musicians across various genres and roles—composers, producers, and performers—to provide nuanced, expert feedback. This collaboration will allow us to better understand industry-specific needs and fine-tune HarmonyCloak to ensure it aligns with the rigorous standards of the music industry. By integrating their expertise, we aim to make HarmonyCloak more robust, practical, and suitable for professional use, ensuring both aesthetic and technical integrity are preserved in music while protecting against unauthorized AI exploitation.

Broadening Testing to Multiple Compression Formats and Platforms. While HarmonyCloak has proven effective against MP3 compression, the real-world use of digital music involves a broader range of compression formats, especially as used by major streaming platforms such as YouTube, Spotify, and SoundCloud. These platforms employ proprietary algorithms and different bitrate settings that could interact unpredictably with the defensive noise applied by HarmonyCloak. To ensure the method's reliability across various listening environments, we will expand testing to cover a wider range of compression formats and streaming technologies. The goal is to enhance HarmonyCloak 's robustness across all major digital platforms, ensuring that protective measures remain effective regardless of how the music is distributed.

Ensuring Long-Term Effectiveness Against Evolving AI Technologies. One of the key challenges for HarmonyCloak is maintaining its long-term effectiveness in the face of rapidly evolving AI models. Current techniques for cloaking music from generative AI may become less effective as new AI advancements and attack strategies emerge. To address this, future work will focus on strengthening the robustness of perturbation-based unlearnable examples in music. Drawing on lessons from the image domain, where new attacks have succeeded in bypassing similar defenses (e.g. [89], [90]), we will explore these challenges in the music domain and continuously adapt our methods. This will ensure sustained protection of musicians' rights and creative works against unauthorized exploitation by increasingly sophisticated AI technologies.

10. Conclusion

In this paper, we addressed the growing concerns regarding the unauthorized exploitation of musicians' music by generative AI models. We propose a defensive framework, HARMONYCLOAK, that leverages imperceptible noise to safeguard music from generative AI models. By introducing noise that disrupts key musical characteristics, we force the generative network to deviate from the training music, thus minimizing the risk of knowledge transfer. Our extensive experiments and evaluations demonstrate the effectiveness of HARMONYCLOAK in both white-box and black-box settings, highlighting its robustness and practicality. Furthermore, through this research, we contribute valuable insights and guidance on data unlearnability for generative models, extending its applicability beyond music protection.

11. Acknowledgement

We would like to thank our anonymous reviewers and shepherd for their insightful feedback. This work is supported in part by NSF CNS-2114161, ECCS-2132106, CBET-2130643, CNS-2403529, CRII-2246067, ATD-2427915, and POSE-2346158.

References

[1] Y. Cao, S. Li, Y. Liu, Z. Yan, Y. Dai, P. S. Yu, and L. Sun, "A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt," arXiv preprint arXiv:2303.04226, 2023.

[2] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in International Conference on Machine Learning, pp. 8821–8831, PMLR, 2021.

[3] OpenAI, "Introducing chatgpt." https://openai.com/blog/chatgpt, 2023.

[4] M. Chui, E. Hazan, R. Roberts, A. Singla, K. Smaje, A. Sukharevsky, Y. Lareina, and R. Zemmel, "The economic potential of generative ai: The next productivity frontier," McKinsey Global Institute, vol. 2, 2023.

[5] M. Civit, J. Civit-Masot, F. Cuadrado, and M. J. Escalona, "A systematic review of artificial intelligence-based music generation: Scope, applications, and future trends," Expert Systems with Applications, p. 118190, 2022.

[6] "Beethoven's last symphony finished by AI." https://www.dw.com/en/beethovens-last-symphony-finished-by-ai/a-59412362, 2021 (Accessed 30-Apr-2023).

[7] "AI and music." https://businesspostbd.com/opinion-todays-paper/ai-and-music-32167, 2021 (Accessed 30-Apr-2023).

[8] T. B. Burnett and J. Taplin, "Opinion — to protect human artistry from ai, new safeguards might be essential," Mar 2023.

[9] V. Kennedy, "The rise of ai and the impact it could have on the music industry." https://cointelegraph.com/news/the-rise-of-ai-and/the-impact-it-could-have-on-the-music-industry, 2023.

[10] S. Shan, E. Wenger, J. Zhang, H. Li, H. Zheng, and B. Y. Zhao, "Fawkes: Protecting privacy against unauthorized deep learning models," in Proceedings of the 29th USENIX Security Symposium, 2020.

[11] H. Huang, X. Ma, S. M. Erfani, J. Bailey, and Y. Wang, "Unlearnable examples: Making personal data unexploitable," arXiv preprint arXiv:2101.04898, 2021.

[12] S. Fu, F. He, Y. Liu, L. Shen, and D. Tao, "Robust unlearnable examples: Protecting data against adversarial learning," arXiv preprint arXiv:2203.14533, 2022.

[13] J. Ren, H. Xu, Y. Wan, X. Ma, L. Sun, and J. Tang, "Transferable unlearnable examples," arXiv preprint arXiv:2210.10114, 2022.

[14] Y. Yang, Z. Zhang, G. Miklau, M. Winslett, and X. Xiao, "Differential privacy in data publication and analysis," in Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pp. 601–606, 2012.

[15] C. Gentry, "Fully homomorphic encryption using ideal lattices," in Proceedings of the forty-first annual ACM symposium on Theory of computing, pp. 169–178, 2009.

[16] B. Pinkas, "Cryptographic techniques for privacy-preserving data mining," ACM Sigkdd Explorations Newsletter, vol. 4, no. 2, pp. 12–19, 2002.

[17] S. Shan, J. Cryan, E. Wenger, H. Zheng, R. Hanocka, and B. Y. Zhao, "Glaze: Protecting artists from style mimicry by text-to-image models," arXiv preprint arXiv:2302.04222, 2023.

[18] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, 2018.

[19] J. Liu, Y. Dong, Z. Cheng, X. Zhang, X. Li, F. Yu, and M. Sun, "Symphony generation with permutation invariant language model," arXiv preprint arXiv:2205.05448, 2022.

[20] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, et al., "Musiclm: Generating music from text," arXiv preprint arXiv:2301.11325, 2023.

[21] 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.

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

[23] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in 2017 ieee symposium on security and privacy (sp), pp. 39–57, Ieee, 2017.

[24] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," arXiv preprint arXiv:1706.06083, 2017.

[25] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, "Recent advances in adversarial training for adversarial robustness," arXiv preprint arXiv:2102.01356, 2021.

[26] A. E. Cinà, K. Grosse, A. Demontis, S. Vascon, W. Zellinger, B. A. Moser, A. Oprea, B. Biggio, M. Pelillo, and F. Roli, "Wild patterns reloaded: A survey of machine learning security against training data poisoning," ACM Computing Surveys, vol. 55, no. 13s, pp. 1–39, 2023.

[27] A. Saha, A. Subramanya, and H. Pirsiavash, "Hidden trigger backdoor attacks," in Proceedings of the AAAI conference on artificial intelligence, vol. 34, pp. 11957–11965, 2020.

[28] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," arXiv preprint arXiv:1712.05526, 2017.

[29] L. Fowl, M. Goldblum, P.-y. Chiang, J. Geiping, W. Czaja, and T. Goldstein, "Adversarial examples make strong poisons," Advances in Neural Information Processing Systems, vol. 34, pp. 30339–30351, 2021.

[30] J. Rothstein, MIDI: A comprehensive introduction, vol. 7. AR Editions, Inc., 1992.

[31] "Midi, daws, and virtual instruments explained." https://www.fortecomposeracademy.com/blog/midi-daws-and-virtual-instruments-explained, 2019 (Accessed 4-Mar-2024).

[32] Q. Huang, A. Jansen, J. Lee, R. Ganti, J. Y. Li, and D. P. Ellis, "Mulan: A joint embedding of music audio and natural language," arXiv preprint arXiv:2208.12415, 2022.

[33] J. Lan, R. Zhang, Z. Yan, J. Wang, Y. Chen, and R. Hou, "Adversarial attacks and defenses in speaker recognition systems: A survey," Journal of Systems Architecture, vol. 127, p. 102526, 2022.

[34] B. Yan, J. Lan, and Z. Yan, "Backdoor attacks against voice recognition systems: A survey," arXiv preprint arXiv:2307.13643, 2023.

[35] Z. Li, Y. Wu, J. Liu, Y. Chen, and B. Yuan, "Advpulse: Universal, synchronization-free, and targeted audio adversarial attacks via subsecond perturbations," in Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS), pp. 1121–1134, 2020.

[36] C. Shi, T. Zhang, Z. Li, H. Phan, T. Zhao, Y. Wang, J. Liu, B. Yuan, and Y. Chen, "Audio-domain position-independent backdoor attack via unnoticeable triggers," in Proceedings of the 28th Annual International Conference on Mobile Computing And Networking (MobiCom), pp. 583–595, 2022.

[37] Y. Qin, N. Carlini, G. Cottrell, I. Goodfellow, and C. Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," in International conference on machine learning, pp. 5231–5240, PMLR, 2019.

[38] L. Schönherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, "Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding," arXiv preprint arXiv:1808.05665, 2018.

[39] R. Duan, Z. Qu, S. Zhao, L. Ding, Y. Liu, and Z. Lu, "Perception-aware attack: Creating adversarial music via reverse-engineering human perception," in Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, pp. 905–919, 2022.

[40] N. Sankaran, W. F. Thompson, S. Carlile, and T. A. Carlson, "Decoding the dynamic representation of musical pitch from human brain activity," Scientific reports, vol. 8, no. 1, p. 839, 2018.

[41] G. F. Welch, M. Biasutti, J. MacRitchie, G. E. McPherson, and E. Himonides, "The impact of music on human development and well-being," 2020.

[42] E. Zwicker and H. Fastl, Psychoacoustics: Facts and models, vol. 22. Springer Science & Business Media, 2013.

[43] B. C. Moore, An introduction to the psychology of hearing. Brill, 2012.

[44] International Organization for Standardization, "Information technology – coding of moving pictures and associated audio for digital storage media at up to 1.5 mbits/s – part 3: Audio," Standard ISO 11172-3, International Organization for Standardization, 1993.

[45] D. Eck and J. Schmidhuber, "Finding temporal structure in music: Blues improvisation with lstm recurrent networks," in Proceedings of the 12th IEEE workshop on neural networks for signal processing, pp. 747–756, IEEE, 2002.

[46] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," arXiv preprint arXiv:1206.6392, 2012.

[47] G. Hadjeres and F. Nielsen, "Interactive music generation with positional constraints using anticipation-rnns," arXiv preprint arXiv:1709.06404, 2017.

[48] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, "Midinet: A convolutional generative adversarial network for symbolic-domain music generation," arXiv preprint arXiv:1703.10847, 2017.

[49] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient," in Proceedings of the AAAI conference on artificial intelligence, vol. 31, 2017.

[50] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer," arXiv preprint arXiv:1809.04281, 2018.

[51] C. Donahue, H. H. Mao, Y. E. Li, G. W. Cottrell, and J. McAuley, "Lakhnes: Improving multi-instrumental music generation with cross-domain pre-training," arXiv preprint arXiv:1907.04868, 2019.

[52] M. Zeng, X. Tan, R. Wang, Z. Ju, T. Qin, and T.-Y. Liu, "Musicbert: Symbolic music understanding with large-scale pre-training," arXiv preprint arXiv:2106.05630, 2021.

[53] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," arXiv preprint arXiv:2306.05284, 2023.

[54] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 30, pp. 495–507, 2021.

[55] Y.-A. Chung, Y. Zhang, W. Han, C.-C. Chiu, J. Qin, R. Pang, and Y. Wu, "W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training," in 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 244–250, IEEE, 2021.

[56] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," arXiv preprint arXiv:2210.13438, 2022.

[57] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," Communications of the ACM, vol. 63, no. 11, pp. 139–144, 2020.

[58] R. B. Dannenberg, "The interpretation of midi velocity," in ICMC, 2006.

[59] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," stat, vol. 1050, no. 9, 2017.

[60] N. Ketkar and N. Ketkar, "Stochastic gradient descent," Deep learning with Python: A hands-on introduction, pp. 113–132, 2017.

[61] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," arXiv preprint arXiv:1803.02999, 2018.

[62] C. Harte, M. Sandler, and M. Gasser, "Detecting harmonic change in musical audio," in Proceedings of the 1st ACM workshop on Audio and music computing multimedia, pp. 21–26, 2006.

[63] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms.," in INTERSPEECH, pp. 2350–2354, 2019.

[64] C. Raffel, Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching. 331 Ph. D. PhD thesis, thesis, Columbia University, 2016.

[65] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," arXiv preprint arXiv:1711.05101, 2017.

[66] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, et al., "Cnn architectures for large-scale audio classification," in 2017 ieee international conference on acoustics, speech and signal processing (icassp), pp. 131–135, IEEE, 2017.

[67] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "Youtube-8m: A large-scale video classification benchmark," arXiv preprint arXiv:1609.08675, 2016.

[68] "Harmonycloak project website." https://mosis.eecs.utk.edu/harmonycloak.html, 2024.

[69] R. Martin, "Spectral subtraction based on minimum statistics," power, vol. 6, no. 8, pp. 1182–1185, 1994.

[70] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," Nature, vol. 401, no. 6755, pp. 788–791, 1999.

[71] Y. Yang, T. Y. Liu, and B. Mirzasoleiman, "Not all poisons are created equal: Robust training against data poisoning," in International Conference on Machine Learning, pp. 25154–25165, PMLR, 2022.

[72] E. Borgnia, J. Geiping, V. Cherepanova, L. Fowl, A. Gupta, A. Ghiasi, F. Huang, M. Goldblum, and T. Goldstein, "Dp-instahide: Provably defusing poisoning and backdoor attacks with differentially private data augmentations," arXiv preprint arXiv:2103.02079, 2021.

[73] L. A. Hiller Jr and L. M. Isaacson, "Musical composition with a high-speed digital computer," Journal of the Audio Engineering Society, vol. 6, no. 3, pp. 154–160, 1958.

[74] G. Nierhaus, Algorithmic composition: paradigms of automated music generation. Springer Science & Business Media, 2009.

[75] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," in International conference on machine learning, pp. 4364–4373, PMLR, 2018.

[76] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," arXiv preprint arXiv:2005.00341, 2020.

[77] "Mubert-inc.." https://github.com/MubertAI/Mubert-Text-to-Music, 2022, 20222.

[78] S. Forsgren and H. Martiros, "Riffusion - Stable diffusion for real-time music generation," 2022.

[79] D. Yu, H. Zhang, W. Chen, J. Yin, and T.-Y. Liu, "Indiscriminate poisoning attacks are shortcuts," 2021.

[80] Z. Liu, Z. Zhao, A. Kolmus, T. Berns, T. van Laarhoven, T. Heskes, and M. Larson, "Going grayscale: The road to understanding and improving unlearnable examples," arXiv preprint arXiv:2111.13244, 2021.

[81] Z. Zhao, J. Duan, X. Hu, K. Xu, C. Wang, R. Zhang, Z. Du, Q. Guo, and Y. Chen, "Unlearnable examples for diffusion models: Protect data from unauthorized exploitation," arXiv preprint arXiv:2306.01902, 2023.

[82] J. Zhang, X. Ma, Q. Yi, J. Sang, Y.-G. Jiang, Y. Wang, and C. Xu, "Unlearnable clusters: Towards label-agnostic unlearnable examples," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3984–3993, 2023.

[83] Y. Liu, K. Xu, X. Chen, and L. Sun, "Stable unlearnable example: Enhancing the robustness of unlearnable examples via stable error-minimizing noise," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 3783–3791, 2024.

[84] J. Ye and X. Wang, "Ungeneralizable examples," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11944–11953, 2024.

[85] R. Chen, H. Jin, J. Chen, and L. Sun, "Editshield: Protecting unauthorized image editing by instruction-guided diffusion models," arXiv preprint arXiv:2311.12066, 2023.

[86] Y. Liu, C. Fan, Y. Dai, X. Chen, P. Zhou, and L. Sun, "Metacloak: Preventing unauthorized subject-driven text-to-image diffusion-based synthesis via meta-learning," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 24219–24228, 2024.

[87] S. Shan, W. Ding, J. Passananti, S. Wu, H. Zheng, and B. Y. Zhao, "Nightshade: Prompt-specific poisoning attacks on text-to-image generative models," in 2024 IEEE Symposium on Security and Privacy (SP), pp. 212–212, IEEE Computer Society, 2024.

[88] Z. Yu, S. Zhai, and N. Zhang, "Antifake: Using adversarial audio to prevent unauthorized speech synthesis," in Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, pp. 460–474, 2023.

[89] P. Sandoval-Segura, V. Singla, J. Geiping, M. Goldblum, and T. Goldstein, "What can we learn from unlearnable datasets?," Advances in Neural Information Processing Systems, vol. 36, 2024.

[90] T. Qin, X. Gao, J. Zhao, K. Ye, and C.-Z. Xu, "Learning the unlearnable: Adversarial augmentations suppress unlearnable example attacks," arXiv preprint arXiv:2303.15127, 2023.

TABLE 10: Comparison of the quality of generated music between HarmonyCloak and random Gaussian noise.

Training	Noise Budget	FAD
Clean Music	-	1.53
Random Gaussian Noise	0.1	1.59
	0.2	4.9
	0.3	9.4
Unlearnable Music	-	11.3

Appendix A. Appendix

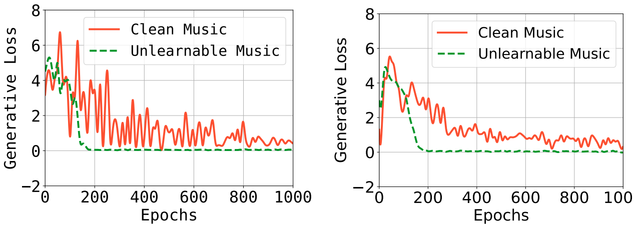
A.1. Additional Experimental Settings

For MuseGAN, we employ both the composer model, which generates music from scratch, and the conditional model, which allows for a track-conditional generation. The composer model enables us to explore the generation of music without any specific conditioning input. In contrast, the conditional model inputs piano tracks and generates the remaining four tracks. Regarding SymphonyNet, we configure the model with an embedding size of 512 for event tokens, durations, instruments, and 3-D positional embedding. The event token vocabulary was derived using the Music BPE algorithm [19], resulting in a final vocabulary size of 1,000. The vocabulary sizes for durations, instruments, and 3-D positional embeddings were determined based on the dataset's characteristics. In SymphonyNet, the linear transformer decoder consists of 12 self-attention layers, each comprising 16 attention heads. For MusicLM, as it works on waveform-based music and our method is specialized in MIDI format, we add an extra layer of midi-wav conversion on each round of the training. We use pre-trained SoundStream [54] and w2v-BERT [55] for extracting acoustic and semantic tokens while training MuLan [32]. For generating unlearnable music, we use the MuLan loss as the generative loss.

A.2. Comparison with Random Gaussian Noise

To understand the true potential of HarmonyCloak, we set out to assess the quality of generated music when trained on music samples perturbed by varying levels of random Gaussian noise. The results, presented in Table 10, exhibit the average FAD scores of the generated music across different noise budgets. In this context, the term "noise budget" refers to the ratio of added Gaussian noise's power to that of the original music signal.

When we introduce minimal Gaussian noise (noise budget is set to 0.1), we find that the quality of the generated music (FAD score) remains comparable to the clean music, showcasing the generative models' resilience to subtle training data noise. However, as the noise budget of random Gaussian noise increases, we observe a notable decrease in the perceptual quality of the training samples, reflected by a corresponding rise in the FAD score for the generated



(a) Untrained MusicLM (b) Pre-trained MusicLM

Figure 11: The training generative loss curves of HarmonyCloak.

music. In contrast, when the model is trained on the unlearnable music generated by HarmonyCloak, the generated music exhibits a significantly higher FAD score, reaching 11.3. This demonstrates that our approach offers greater unlearnability compared to music examples perturbed by random noise, even when the noise budget is increased to 0.3. Notably, the noise introduced by HarmonyCloak remains imperceptible, thus HarmonyCloak achieves this unlearnability without compromising the listening experience of the training music sample.

A.3. Model Fine-tuning with Unlearnable Music

Our results, presented in the main paper, are based on training generative AI models from scratch using a mixed dataset containing both clean and unlearnable music tracks. However, it is also possible for generative AI models to be pre-trained on millions of music samples. These pre-trained models can then be fine-tuned with new music to generate similar styles. To evaluate the effectiveness of our method in this context, we conducted experiments generating unlearnable music in the white-box setting using a pre-trained MusicLM model. We then fine-tuned the model on these unlearnable tracks. Our findings revealed that generating unlearnable music for the pre-trained MusicLM model introduced a higher level of noise and posed greater challenges in cloaking features the model had previously learned. In Figure 11, we compare the performance of HarmonyCloak when fine-tuning the pre-trained model to its performance when training a model from scratch. Importantly, we found that generating unlearnable music for pre-trained MusicLM models, to achieve results comparable to the scratch-trained model required 53% more iterations (inner optimization in Equation 5).

TABLE 12: Time cost analysis of generating unlearnable music.

Method	Time for Fixed Length Sample (sec)	
	white-box	black-box
Random Gaussian Noise	0.0034	0.0037
L2 norm	7.1	13.2
$L\infty$	7.3	13.1
HarmonyCloak	12.5	25.7

A.4. Time Complexity Analysis

We have conducted time cost analysis for generating various types of defensive noises in both white-box and black-box settings on four Nvidia A100 GPUs, and the results are presented in Table 12. We choose a 25 second instrumental piece for the experiments. We use MuseGAN as the target model. From the table, we observe that HarmonyCloak, while efficient, does take slightly more time compared to the L_p-norm-based method due to the calculation of psychoacoustic features. In the white-box setting, generating an unlearnable music sample requires only 12.5 seconds, while in the black-box setting it takes 25.7 seconds. This demonstrates HarmonyCloak's efficiency and potential for large-scale deployment, with remarkable speed that makes it well-suited for handling substantial workloads and real-world applications.

A.5. Model Transferability Analysis

We also evaluated the transferability of the unlearnable music generated by HarmonyCloak in the white-box setting. In this context, a distinct model is employed to generate unlearnable music, differing from the original training model. To perform this, we produced unlearnable music with MuseGAN and employed it to train the SymphonyNet model. We conducted a temporal analysis and the results of this analysis are presented in Table 11, which illustrates the temporal characteristics of the music generated by SymphonyNet. The results reveal that unlearnable music exhibits a significantly lower percentage of EB compared to models trained on clean music and even the original training data. This observation implies that the generated music lacks rhythm and structure. Furthermore, the higher values of the UPC metric in the models trained with unlearnable music indicate the utilization of nearly all pitch classes in each instrument. Consequently, the music produced will exhibit implausible characteristics to human listeners, affirming that unlearnable music possesses some degree of transferability among distinct models.

TABLE 11: Intra-track performance evaluation of HarmonyCloak trained on transferred unlearnable samples.

Training Music	Empty Bars (EB;%)					Used Pitch Classes (UPC)				Qualified Notes (QN;%)				Drum Pattern (DP;%)
	B	D	G	P	S	B	G	p	S	B	G	P	S	
Clean Music	7.29	8.32	19.7	22.2	10.6	1.59	4.75	4.13	4.22	81.9	72.0	71.1	81.5	88.0
Unlearnable Music	1.09	2.45	1.65	0.96	0.74	11.3	10.1	10.5	10.3	56.3	59.1	68.3	72.2	66.7

Appendix B. Meta-Review

The following meta-review was prepared by the program committee for the 2025 IEEE Symposium on Security and Privacy (S&P) as part of the review process as detailed in the call for papers.

B.1. Summary

The paper presents "HarmonyCloak," a tool designed to protect the intellectual property of instrumental musicians in the age of generative AI. By introducing subtle, imperceptible changes to audio waveforms, HarmonyCloak prevents modern generative models from learning meaningful information from the music.

B.2. Scientific Contributions

- Creates a New Tool to Enable Future Science
- Establishes a New Research Direction

B.3. Reasons for Acceptance

1) This paper creates a new tool to enable future science. The use of subtle perturbations to make music audio unlearnable is both interesting and new. As one of the first works in this space, the paper has the potential to provide a building block and stimulate follow-up works in this important direction.
2) This paper considers an important and timely issue of protecting human musicians from unauthorized model training.
3) The authors tested HarmonyCloak across a wide range of scenarios, including various rhythmic structures and instrumentations, different levels of access to generative models (both white-box and black-box), and robustness against audio processing techniques such as compression. These extensive experiments demonstrate the effectiveness of HarmonyCloak's protection.