MARingBA: Music-Adaptive Ringtones for Blended Audio Notification Delivery

ANONYMOUS AUTHOR(S)

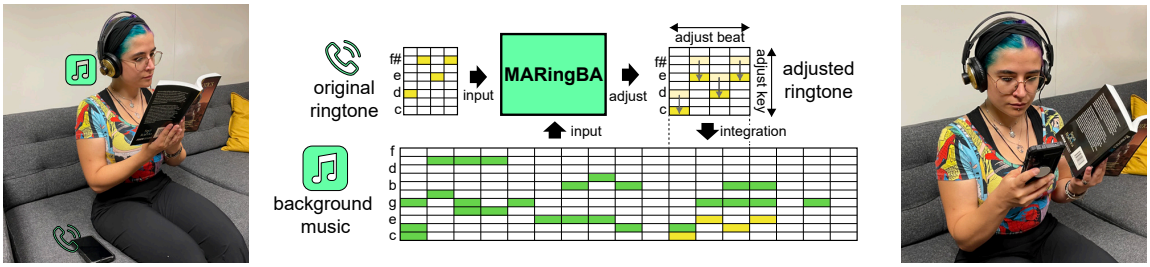

Fig. 1. We present MARingBA, an approach that automatically blends audio notifications into the music users' are listening to. The user is reading a book when an audio notification arrives. Our system adapts the notification to match the beat and key of the background music, to create a less disruptive experience.

Audio notifications provide users with an efficient way to access information beyond their current focus of attention. Current notification delivery methods, like phone ringtones, are primarily optimized for high noticeability, enhancing situational awareness in some scenarios but causing disruption and annoyance in others. In this work, we build on the observation that music listening is now a commonplace practice and present MARingBA, a novel approach that blends ringtones into background music to modulate their noticeability. We contribute a design space exploration of music-adaptive manipulation parameters, including beat matching, key matching, and timbre modifications, to tailor ringtones to different songs. Through two studies, we demonstrate that MARingBA supports content creators in authoring audio notifications that fit low, medium, and high levels of urgency and noticeability. Additionally, end users express a preference for music-adaptive audio notifications over conventional delivery methods, such as volume fading.

CCS Concepts: • **Human-centered computing** → **Auditory feedback**; **Sound-based input / output**; • **Applied computing** → **Sound and music computing**.

ACM Reference Format:

Anonymous Author(s). 2024. MARingBA: Music-Adaptive Ringtones for Blended Audio Notification Delivery. In . ACM, New York, NY, USA, 23 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Audio notifications, like ringtones, play a crucial role in how users receive information and are widely employed by modern digital devices, especially mobile phones, to relay time-sensitive updates, including incoming calls, upcoming calendar events, or messages. Presently, most auditory notifications are intentionally designed to be highly noticeable, ensuring users don't miss them. While this design effectively captures the user's attention, it can also lead to disruptions in various contexts.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2024 Association for Computing Machinery.

Manuscript submitted to ACM

Consider this scenario: a user is engrossed in a book while enjoying their favorite song (Figure 1). In the event of a notification, most digital devices will automatically lower the background music volume and deliver the alert audibly. While this approach guarantees that the user detects the notification, it often proves distracting, disruptive, and overall detrimental to the user's music-listening experience. This is particularly the case for non-urgent notifications that require a timely but not necessarily immediate response by users.

In the past, researchers have tried to address this challenge through methods like substituting notifications with audio effects [3, 7], integrating music snippets in pre-composed music soundscapes [14, 32], and embedding ringtone sounds in single-timbre music using timbre transfer techniques [49, 50]. While these prior approaches have shown promise in creating more musically-integrated notifications, they are subject to two significant limitations. First, they often rely on completely replacing notifications that are highly familiar to users, which can diminish recognizability and user comfort. Second, many methods are customized for predefined musical contexts or require songs to be composed with notifications in mind, limiting their adaptability to the diverse music preferences of today's users.

In our work, we present a novel approach that seamlessly integrates audio notifications with users' musical experiences. Drawing inspiration from digital music practices like disc jockeying and remixing, as well as concepts from music information retrieval (MIR), our initial contribution involves exploring a parameter design space of auditory manipulations for creating music-adaptive audio notifications. These parameters, including beat matching, key matching, and timbre modifications, are designed to facilitate the seamless integration of notifications into musical sequences while allowing for customizable degrees of blending.

To validate these parameters and investigate the user experience of a more musically-adaptive approach to delivering audio notifications, we further developed MARingBA, an interactive system enabling real-time manipulation of notifications. MARingBA is designed for content creators and designers of audio notifications. MARingBA incorporates a suite of automated mechanisms for extracting music information and serves as a prototype interface for experimenting with and creating music-adaptive notifications using our design space parameters.

Through two studies, we gather insights on our approach, design space, and system from the perspective of two main stakeholders: *(1)* **content creators** responsible for designing audio notifications, and *(2)* **end-users** who may receive these notifications in the future. Our first design study with six music experts revealed that our design space is highly expressive and enabled them to tailor notification designs to a diverse set of contexts. They were notably able to use MARingBA to blend the notifications they were given with multiple songs and to accommodate for various noticeability and urgency requirements (e. g., designing for casual weather alerts versus work scenarios requiring an immediate response). In a second experiment with end-users, we validated that our parameters could modulate the noticeability of audio notifications while producing a preferred user experience to standard notification delivery mechanisms.

In summary, we make the following contributions:

- A novel design space of parameters for adapting notifications to a background musical context in a harmonic manner, which also enables the modulation of its noticeability,
- MARingBA, a novel system that implements our design space parameters for authoring music-adaptive notifications,
- Insights from a design study with *content creators* ($n = 6$) on the utility of our parameters and system for creating music-adaptive notifications,
- Results from an empirical study with *end users* ($n = 12$) showing that music-adaptive audio notifications are preferred over a standard volume-fading baseline, while exhibiting controllable detection rates.

2 RELATED WORK

Our work builds on relevant related work from the fields of digital notifications, audio notifications, and music information retrieval.

2.1 Digital Notifications

Notifications are a ubiquitous feature of modern digital devices and a longstanding topic of interest within HCI research [35, 44]. By proactively delivering visual, haptic, or auditory alerts, notifications serve as an efficient way to convey information to users from sources outside their primary focus of attention [41, 44]. Early research has shown that notifications may benefit users' informational awareness [41]; however, if presented at an inopportune time or at an inappropriate frequency, notifications become a source of disruption and annoyance [1, 6, 11]. Disruptions from such notifications lead to errors during task performance [1], anxiety [6], and productivity loss [11]. As the proliferation of digital interfaces generates an ever-increasing volume of notifications, HCI researchers have continued to study and improve notifications in various devices (e. g., multi- and cross-device ecosystems [16, 47], smart homes and intelligent living environments [34, 46], virtual reality [23]).

One substantial subset of existing literature has investigated *when* notifications should optimally be delivered. For instance, early work by Czerwinski et al. [18] and Horvitz [27] found that the disruptiveness of notifications depends on their contents, the nature of the task that the user is engaged in, as well as the user's level of engagement. Related research also found that scheduling notifications at natural task *breakpoints* reduces the cost of interruption [2, 5] Consequently, Bailey and Konstan [5] and Iqbal and Bailey [29] have suggested that it may be valuable to imbue devices with some level of awareness of the user's task structures and attention. Iqbal and Bailey [30] implemented a computational approach that operationalizes these insights in the desktop domain. Hudson et al.'s Wizard of Oz study demonstrated the potential value of making such predictions about interruptibility through sensors [28].

In prior research, there is also a complimentary line of work investigating *how* notifications should be designed. Arroyo et al. [4], for instance, found an interaction effect between the modality in which a notification was presented and participants' prior experiences on its effectiveness. In the desktop domain, Müller et al. [42] found that users' ability to detect a notification depends on its background and placement on the screen. Prior research generally agrees that the noticeability or attentional draw of notifications should be proportional to their utility [24, 37–39, 43].

Our work primarily aims to innovate how notifications can be delivered. Drawing inspiration from Gluck et al. [24] in particular, we introduce an approach to modulating the attentional draw of notifications by embedding it within background music to varying degrees. Our approach intends to serve as a foundation for adaptively curating notifications based on their potential utility to the end-user in future applications.

2.2 Audio Notifications

In the realm of auditory information presentation, two prominent methods have emerged: auditory icons as proposed by Gaver [22], and Earcons as introduced by Blattner et al. [9]. Auditory icons leverage real-world sounds that correspond to their virtual function, further conveying multi-dimensional data by modulating various sound qualities. In contrast, Earcons consist of composed sequences with no inherent association to their representation, necessitating users to learn the connection.

Both Earcons and Auditory icons have been explored as means of delivering notifications to users via mobile devices [21]. They have demonstrated efficacy in critical contexts [25] as well as in enhancing task performance [36].

Nevertheless, while audio notifications can effectively alert users, they may also introduce irritations and disruptions [13]. In our work, we aim to alleviate the disruption induced by audio notifications through their integration with musical elements.

Prior work aimed to make audio notifications less intrusive with various methods. Jung and Butz [14, 32] modified pre-composed music to alert users about incoming information. Their modification included adding or omitting specific instruments to deliver target notifications. Their approach requires music to be composed with the notifications in mind, while ours generalizes to more music. Ananthabhotla and Paradiso [3] substitute audio notification content with audio manipulation on the music itself in their SoundSignaling system to deliver audio notifications. They introduce subtle signals that rely on users' familiarity with a song to detect a notification. Since the manipulations are very subtle (e. g., temporarily changing the rhythm of a song slightly), they are limited in the types of notifications they can deliver. Finally, Yang et al. [49, 50] used timbre transfer to embed ringtones into music. However, their approach still follows a conventional delivery format where the music track itself is faded out, or muted, during the ringtone notification. Their approach does not consider other essential manipulations such as key matching or beat matching to integrate a notification better into music. Our approach takes timbre transfer as one building block in the larger design space of music-adaptive audio notifications.

Besides audio notification, Barrington et al. [7] leverage audio notification as standalone ambient displays, and manipulate an existing music track to communicate human affect state. Lastly, Kari et al. [33] modified songs to align with the affordances of car rides. The manipulations of both approaches serve as inspiration for how audio is processed in MARingBA.

2.3 Sound and Music Computing

Our work relies on techniques from Music Information Retrieval (MIR), particularly for deriving quantifiable features from audio signals. Within MIR, audio is manipulated for creating DJ systems [31], mash-up systems [19], and automatic music mixing systems [45] that are comparable to human experts in each realm. By finding the beat, key, downbeat, and structural segments of each individual song, these automatic systems are capable of making the necessary audio manipulations and selection of pre-existing music tracks to create seamless mixes. While building a system for integrating ringtone snippets into mainstream music involves similar techniques, the nature of common ringtones often being short and monophonic bypasses some challenges faced by automated DJ systems. We leverage the technique from MIR for parameterizing ringtones according to urgency, for example, and delivering them to users in a way that is non-intrusive but still timely.

3 BACKGROUND

In the following, we will first provide background information on several fundamental music concepts relevant to our work. This section can be skipped by knowledgeable readers. For a more comprehensive introduction to music theory and computer music, we refer interested readers to Blatter [8] and Collins [15] respectively. Additionally, we then describe the limitations of current audio notification delivery approaches.

3.1 Relevant music concepts

Music concepts relevant to our work include *tempo*, *pitch*, *note*, and *key*. Throughout the paper, we illustrate these parameters similar to Figure 2. Bars represent individual notes, played in a specific key (y-axis). The notes follow a certain tempo (i. e., beat), which is indicated by the grid cells.

Tempo refers to the speed or pace at which a piece of music is performed, typically measured in beats per minute (BPM). This intuitively corresponds to the rate at which people naturally tap their feet when they listen to music.

Pitch refers to the perceived highness or lowness of a sound, which is determined by the frequency of its vibrations and measured typically in hertz (Hz). Higher frequencies correspond to higher pitches and lower frequencies to lower pitches. In music composition, sounds of different pitches are referred to as *notes*.

A *key* refers to a set of pitches or notes. Songs and musical compositions typically adhere to notes belonging to a single key (e. g., C major). Introducing additional off-key notes typically results in undesirable dissonance (i. e., keys are not in harmony), with notable exceptions in experimental music and jazz, for example, in which dissonance is carefully used as a design element.

Timbre is a broad term used to describe the unique characteristics of a sound that differentiate it from its pitch, and is colloquially described as the "quality of a musical note or sound". It is determined by the combination of overtone frequencies of a sound. An effective way to grasp timbre is by considering, for example, how a guitar and a violin can play the same music at the same pitch and intensity yet sound distinct from each other.

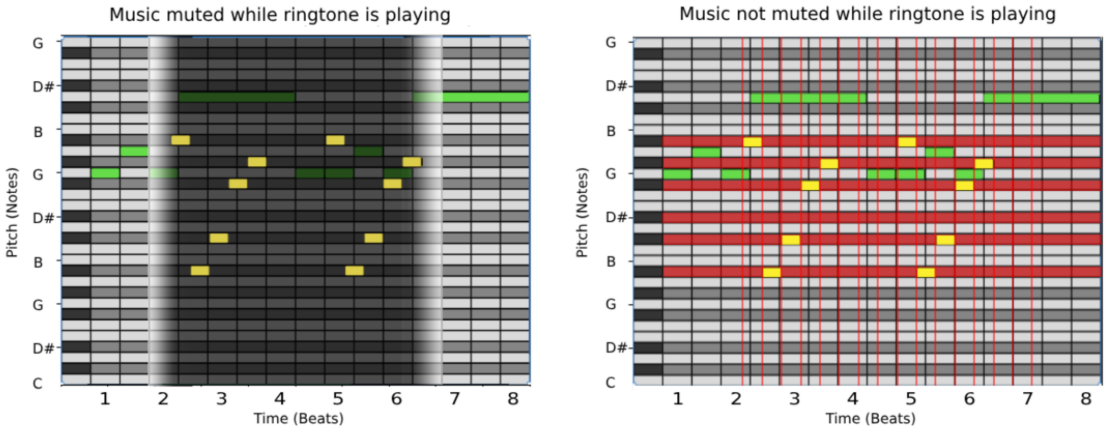

Fig. 2. Left: Integrating a ringtone (yellow) into music (green) by muting the song. The blacked out area indicates a volume decrease. Right: Integrating a ringtone integrated into music without muting the song. Unaligned tempo and notes of dissonance are annotated in red.

3.2 Audio notifications

On current devices, audio notifications are typically delivered in one of two ways: either they *mute* whatever the user is listening to or they are directly *overlaid*. If the user was previously listening to music, the muting approach ensures that the notification is noticed but fully interrupts the user's listening experience.

Alternatively, if the notification is overlaid, users can still hear the music, albeit quieter if its volume is decreased. The sounds from the two sources, however, may clash and result in an unpleasant experience for the human ear. From a musical perspective, this dissonance can be attributed to several factors, such as misalignments in tempo or key, as illustrated in Figure 2. Most music is composed at a consistent tempo, so introducing a rhythmic sound that doesn't align with this tempo, e. g., because it is faster or slower, can be disturbing to the listener. Similarly, when the pitch of the notification doesn't match the music's key, it will be perceived as out of place. Overall, unless an immediate user

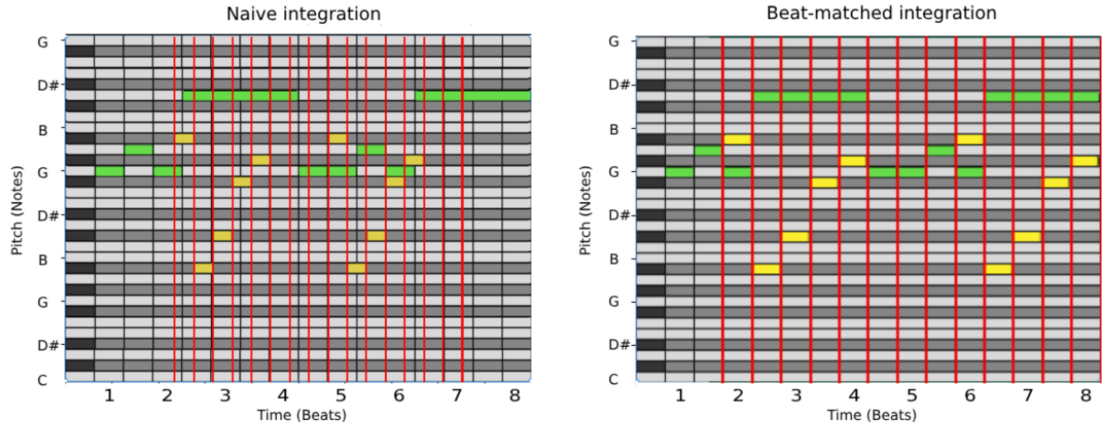

Fig. 3. Left: a naive implementation. The red lines here represent the beat timings of the ringtone and are unaligned with the music. Right: the ringtone is beat-matched by stretching the audio to be played at a slower rate. This perfectly aligns the beats with the music.

response is required, both conventional notification delivery mechanisms—mute and overlay—may lead to a sub-optimal experience, as they do not consider the musical context in which the user is situated.

4 THE MARINGBA PARAMETER SPACE

Our goal is to automatically generate ringtone music blends that resemble the quality of manually crafted mixes by human mash-up artists. To achieve this, we propose an approach centered around defining a set of distinct music feature modification parameters. These parameters are grounded in music theory, and inspired by music practices that involve blending multiple musical audio sequences together, such as DJing or sampling.

In the following, we provide an in-depth description of these parameters, their conceptual implementation, and their role in achieving effective ringtone-music adaptation.

4.1 Beat matching

Beat matching refers to slowing down or speeding up one or both of the clips until their tempo becomes the same. This technique is used by DJs and mash-up artists to align the tempo of two different songs and create a synchronized mix that listeners can dance to. In addition, beat matching refers to manually synchronizing the onset of beats to align across songs.

Assuming the timestamp of every beat in a piece of music is known, e. g., by using rhythm extraction software [12], we first calculate the average interval between all pairwise beats in a song. The average tempo in beats-per-minute (BPM) is then calculated as

$$\text{average tempo (BPM)} = \frac{60}{\text{average interval}} \tag{1}$$

We can then use the tempo estimation to beat-match the notification audio to synchronize with the music by calculating

$$\text{time-stretch amount} = \frac{\text{average tempo music}}{\text{average tempo ringtone}} \tag{2}$$

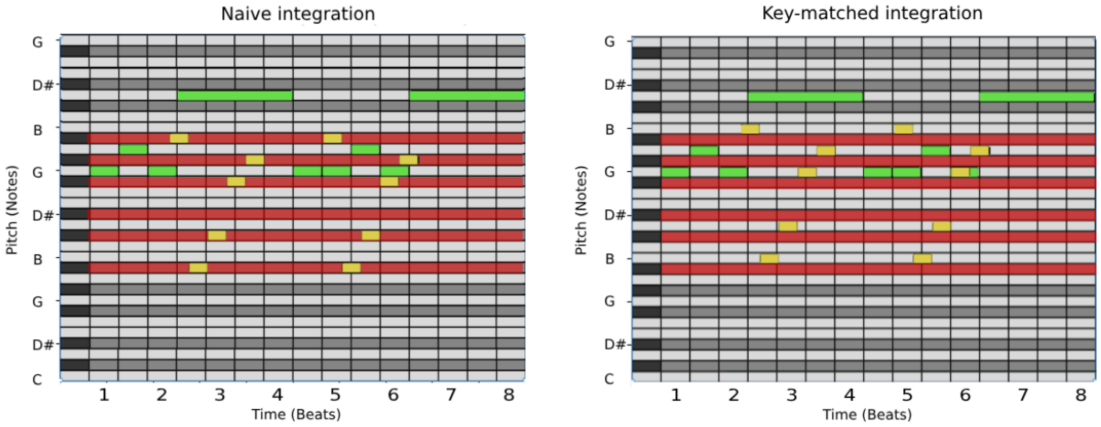

Fig. 4. The horizontal red lines represent the pitches that do not fit in the music's key. Left: a naive implementation. Here, the ringtone pitches do not match the music and create dissonance as a result. Right: the ringtone is key-matched by pitch-shifting the audio to be played at a higher pitch, where the note pitches are perfectly aligned with the key of the music.

The time stretch amount is then applied to the ringtone to match the tempo of the music. Furthermore, the beat onset of the ringtone is aligned with the beat of the music. Figure 3 illustrates a naive implementation with misaligned beats and a beat-matched version.

4.2 Key matching

When pitches outside of the music's key are introduced, dissonant clashes can occur. Key matching, sometimes also referred to as harmonic mixing, is the act of making sure two songs that are being blended share a similar key. One simple way to ensure harmony is to shift the pitch (i. e., frequencies) of the key of the ringtone to match the key of the song. In Western music theory, frequencies are split into 12 equidistant notes, which are the main building blocks for all keys. The distance between subsequent notes is called semitone. Key matching through pitch shifting therefore relates to moving between notes by shifting their frequencies with the correct semitones.

To achieve this, the key-matched frequency is calculated as

$$\text{key-matched frequency} = \text{original frequency} \times 2^{1/12} \times \text{semitones}. \tag{3}$$

As an example, matching a ringtone in the key of B to a song in the key of C, i. e., a difference of one note, the ringtone should be pitch-shifted upwards by 1 semitone. As shown in Figure 4, the ringtone does not use pitches that fit into the key of the music. We rectify this by pitch-shifting the ringtone up by one semitone, avoiding dissonant notes and staying in harmony with the music.

4.3 Scheduling

Conventional notification delivery mechanisms typically alert the user as soon as notifications are received. In a musical context, this may coincide with an undesirable temporal placement where the notification is not aligned with the background rhythm. There are several ways to mitigate this challenge, including waiting with playback until the next beat, next bar, or the start of the next four bars. In music theory, a bar, or measure, is a segment of time that involves the

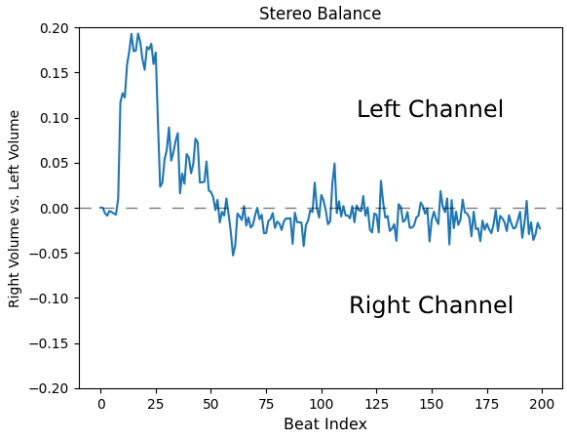

Fig. 5. A plot of the stereo balance of a song. Values above 0.0 relate to the left stereo channel. Values below 0.0 relate to the right channel. If a song is around 0.0, the volume between the two channels is balanced, i. e., both have roughly the same volume. Initially, the song is predominantly panned to the left side, which can be used to integrate a ringtone on the right channel, for example.

grouping of multiple beats, usually four in mainstream music. Structurally, music often has repetitions and variations that happen at the start of every four bars, making those positions suitable candidates to blend ringtones naturally in the structure of the music.

4.4 Panning

Panning refers to how the audio is distributed across different channels in a stereo sound system (e. g., headphones). A sound that is panned to the left will result in more volume from the left speaker, for example. Panning can be used in music-adaptive ringtones to make the sound stand out as coming from a different direction than other sounds presented in the mix of music. In our approach, the ringtone sound is panned from the side which holds less intensity. The song shown in Figure 5, for example, begins with a sequence that is panned to the left side. A ringtone could be panned to the right to counterbalance the difference in volume, and balance the stereo sound.

4.5 Timbre

There exists various way to manipulate the timbre of the music, including instrument transfer, reverb dry/wet level, reverb decay time, and setting high pass and low pass thresholds.

- **Instrument Transfer.** Aside from the original sound of each ringtone, we allow the ringtone's instrument to be replaced with either a piano, violin, or synthesizer, while keeping the same melody and rhythm as the original. Transferring the instrument tone used to play the ringtone can result in a more integrated mix depending on the instrumentation of the song. The original iPhone ringtone, for example, is played on the Marimba, a type of mallet percussion with African origins. This ringtone might not blend well into synth-heavy dance tracks, and would benefit from being transferred to a synthesizer, for example.
- **Reverb.** Reverb refers to the reflections of a sound from the environment it is played in. More spacious environments lead to more reverb signals and longer decay times. Adding simulated reverb on ringtones helps with music integration, as reverb makes the sound more natural by simulating its presence in a physical space.

- **Frequency filtering.** Frequency filtering is the removal of frequency content from a sound that is rich in frequencies. If there were a significant overlap in frequencies between the ringtone and music, the ringtone would be less noticeable. Removing part of the ringtone's frequencies can benefit ringtone integration because it can "free up" room for the song, without significantly impacting the ringtone's volume.

4.6 Volume

Lastly, we can manipulate the volume of the ringtone and music clips in several ways.

- **Adaptive volume.** To make volume settings more generalizable across songs, we can adapt the volume of the ringtone, i. e., increasing the volume to match louder songs, or decreasing the volume to match softer songs.
- **Fade-in.** Fade-in refers to starting at a low volume and gradually transitioning back to a stable level of volume. This transition eases the introduction of the ringtone and makes integration smoother.
- **Ringtone volume.** Adjusting the ringtone's volume directly affects its subtlety when mixed with music. Lowering the volume can make the ringtone blend more smoothly with the music, creating a more seamless and integrated auditory experience.
- **Track specific attenuation.** Furthermore, we also support the adjustment of volume on specific tracks. For example, we can temporarily decrease the volume of the vocals of a song while keeping the accompaniment at full volume. This leaves extra space for the ringtone to be blended in without putting a pause on the flow of music. In contrast to the other parameters, this manipulation targets the music users are listening to.

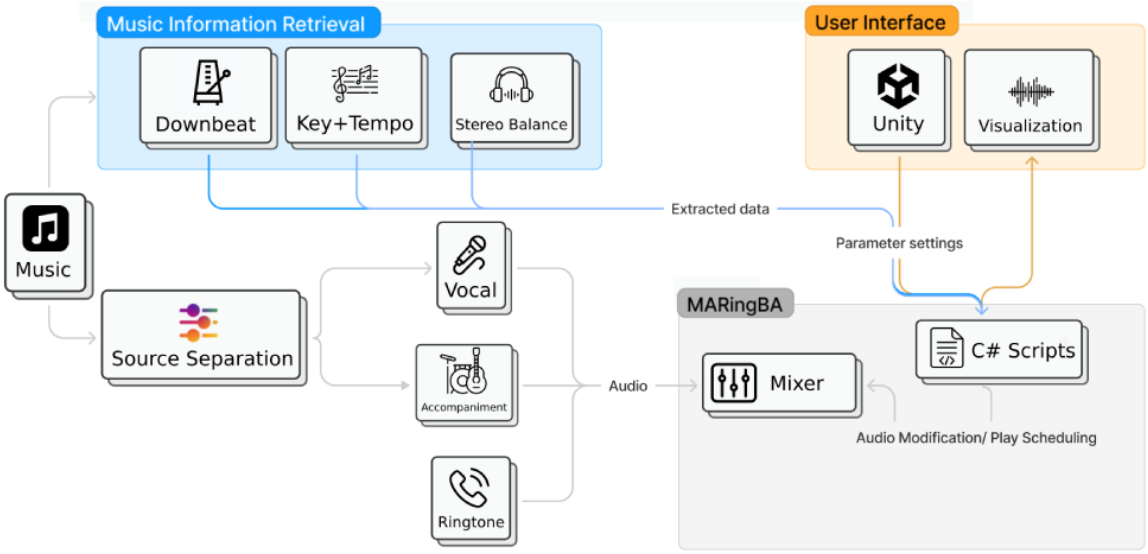

Fig. 6. An overview of the MARingBA system. Music is processed with music information retrieval models to extract downbeat, key, tempo, and stereo balance. The data is then loaded into Unity to support audio modifications.

5 THE MARINGBA SYSTEM

The goal of MARingBA is to create music-adaptive ringtones that blend into the music users are currently listening to. To achieve this, we implemented the adaptation parameters described in section 4. Content creators can leverage those

different manipulations. We demonstrate this in the expert study and validate the designed ringtones in the user study, both described later. In the following, we detail how the individual manipulations are implemented in MARingBA.

Our current implementation includes features for *extracting music information* from the ringtone and background music, and performing *real-time notification feature manipulation.* An overview of the system is illustrated in Figure 6. Apart from information extraction and audio manipulations, MARingBA provides content creators with mechanisms to select target notifications and songs for testing the parameters, playback controls, and manual triggers to simulate the arrival of notifications. The front-end for content creators is implemented in Unity 2021.

5.1 Music Information Extraction

To extract the musical features on which our adaptation parameter manipulations were applied, we pre-processed and extracted relevant information from the input songs and ringtones using Python 3.10 along with several established Music Information Retrieval (MIR) libraries.

We extracted the beat onsets (i. e., the beat start times) using ESSENTIA [12] and downbeats (i. e., the start of bars) using Madmom [10] to enable beat matching and scheduling. To support key matching, we also estimated the key of our input audio clips using ESSENTIA [12]. To support more fine-grained volume adjustments, we isolated the input songs into vocal and instrument tracks with Spleeter [26]. Lastly, to support panning, we computed the left and right channel intensities of our input songs and notifications with Librosa [40].

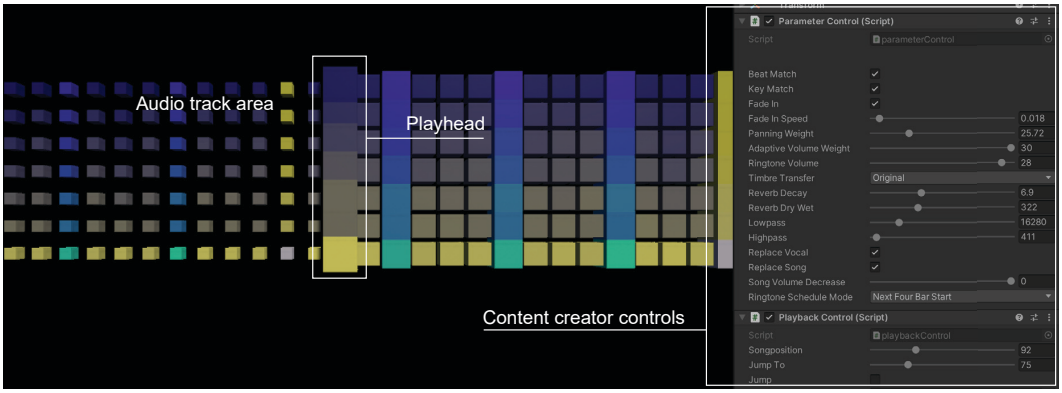

Fig. 7. The MARingBA system interface. The audio track area provides a visualization of the song. Content creators can configure parameters using the control panel on the right.

5.2 Real-time Ringtone Feature Manipulation

Our system interface for performing ringtone feature manipulations is implemented in Unity 2021, shown in Figure 7. The individual controls described in section 5 are implemented as part of a Unity Component exposed in the Editor. The real-time manipulation of ringtone features was implemented using Unity, which provided a versatile platform for audio feature processing. Leveraging Unity's built-in audio engine, we executed essential sound modifications, including play scheduling, volume attenuation, panning adjustments, and pitch stretching. More advanced features such as frequency filtering, reverb dry/wet level, and decay time control were implemented by automating Unity mixer channel settings via C# scripts.

5.2.1 Beat matching. MARingBA estimates the average tempo of the song and time-stretchs the ringtone to match that tempo based on the beat onset information. To mitigate any potential pitch shifts arising from this tempo synchronization (e. g., speeding up a track might increase its pitch), we take advantage of a Unity mixer channel setting. Specifically, we utilized Unity's AudioSource.Pitch feature, which adjusts the speed of the audio clip and the pitch of the audio clip. By configuring the pitch of the mixer channel to 1/ringtone pitch, we counteract any unintended alterations in pitch produced in the previous step. Furthermore, since some songs that are performed live fluctuate slightly in tempo, we dynamically re-align with the beat onsets every time the ringtone repeats itself. Content creators can toggle this feature on or off.

5.2.2 Key matching. Using the key estimation, MARingBA first finds the least amount of pitch shift required to avoid dissonance. It then adjusts the pitch value of the Unity mixer channel to the target pitch. Due to the nature of ringtones being a short melody as opposed to a fully orchestrated song, the pitches used in the ringtone may fit into multiple different keys without causing dissonance. We exploit this quality in our implementation of key matching by iterating through all the different possible keys to shift to and finding the least pitch-shift required to avoid dissonance instead of pitch-shifting to match the exact key the song is in. Content creators can toggle this feature on or off.

5.2.3 Scheduling. Our system enables content creators to define whether a ringtone should be scheduled to the closest start of beat, start of bar, or start of four bars. Timestamps of beats and downbeats (start of bar) are estimated using MIR libraries. We assume that the first downbeat of the song is the start of a section and keep an internal counter to label every fourth beat after that to be the start of four bars. While this assumption worked well for the approximately two dozen popular songs we tested, it might not hold for all music. We hope to implement techniques to overcome this in the future.

5.2.4 Panning. We leverage a beat-by-beat comparison of root-mean-square (RMS) values between the left and right channels to determine if one has a louder volume. Positive values indicate a higher volume on the left channel, and negative values indicate a higher volume on the right channel. MARingBA pans the ringtone sound using Unity's AudioSource.panStereo, setting it to the value of volume difference on the current beat. In our system, content creators control the panning of notifications using a simple slider, with values between 0 and 100. The slider value is a multiplier to the balance data of the current song. A value of 0 will keep the sound of the ringtone centered. If one side is louder at the time a notification is played, MARingBA will play the notification on the other side by multiplying the difference in RMS value with the slider value to determine the volume of the ringtone.

5.2.5 Timbre. Instrument transfer is made possible by pre-rendering the ringtone in different instrument tones (piano, violin, or synthesizer) and selectively unmuting only one instrument track at a time based on the user-selected instrument. Reverb and filtering are both based on Unity's mixer channel effects. The Unity Audio SFX Reverb effect is used for reverb; highpass and lowpass effects are used for frequency filtering. These effects are added to the same mixer channel that the notification is routed to.

In addition to instrument transfer, content creators can control the decay time and dry/wet level of the ringtone. Decay time controls how long it takes for the reflected sound to decrease in intensity, which dictates the perceived space of the reverb simulation. A longer decay time simulates a larger or more reflective space, whereas a shorter decay time simulates a smaller or less reverberant space. Dry/wet level balances the volume between the simulated reflected sound and the original non-reverberant sound.

As final timbre manipulation, content creators can filter frequencies, including high-pass and low-pass cutoffs.

5.2.6 Volume. Notification, accompaniment, and vocals are each routed to their own mixer channel. The volume parameter of each channel are manipulated based on different parameters. Track-specific attenuation settings decrease the volume of specific tracks when the ringtone is played (e. g., decrease volume). Content creators can toggle whether vocals and accompaniment are audible when the ringtone is played. We can further control the rate of gradual fade-in of both the ringtone and the background music by setting a fade-in speed value. When toggled on, the fade-in speed slider (0.01 to 0.2 seconds) can be used to control how long it takes until the ringtone resumes with its original volume. The volume of the ringtone is controlled using sliders, and can be generalized across different songs by enabling the adaptive volume setting. This boosts the volume of the ringtone when the music is loud.

6 NOTIFICATION DESIGN STUDY

To evaluate MARingBA, we first conducted an expert design study. We recruited six musicians and music enthusiasts to define adaptation parameters using our MARingBA system to integrate two NOTIFICATIONS into three SONGS, each time for three different SCENARIOS. We designed the SCENARIOS to involve notifications that demanded varying levels of *urgency* in response, ideally leading to designs with a corresponding attentional draw or noticeability, as proposed in prior work [24]. Our design study ultimately investigates the following questions: (**RQ1**) To what extent do the MARingBA parameters enable content creators to implement sound notifications that adjust their noticeability to contexts with varying levels of urgency? (**RQ2**) What additional parameters need to be supported? Participants reported on their approach and experience while engaging with our system.

6.1 Design

We used a single-variable within-subject design, eliciting adaptation parameters for *low, medium,* and *high* urgency SCENARIOS. In each SCENARIO, each participant designed a set of parameters to integrate two NOTIFICATIONS into three SONGS (3 SCENARIOS $\times$ 2 NOTIFICATIONS = 6 parameter sets in total). The selected NOTIFICATIONS and SONGS participants designed for were set as control factors. NOTIFICATIONS were randomly selected from a set of six and SONGS were randomly selected from a set of 12.

6.1.1 Scenarios. To investigate whether the parameters in MARingBA allowed content creators to customize the noticeability of their notifications for different contexts, we designed three scenarios, each demanding a different level of urgency in response:

- **Low urgency.** A user receives a notification about the temperature on the following day.
- **Medium urgency.** A user receives a reminder about an upcoming meeting in two hours.
- **High urgency.** A user receives an email from their supervisor that requires an immediate response.

6.1.2 Notifications. We curated a set of six notification ringtones from popular mobile devices and applications (e. g., iPhone marimba ringtone, Skype). While we did not explicitly search for notifications that differed in character, our curated set represented a range of tempos (130 − 180 bpm, $M = 158$ bpm, $SD = 22$) and keys (4 represented). The list of notifications can be found in Appendix Section A.1.

6.1.3 Songs. We curated a set of 12 popular songs (with at least 52M views on YouTube) from a range of years (1967 to 2020) and genres (e. g., R&B, hip hop, alternative rock, funk), for example "Happy" by Pharell Williams or "September" by Earth, Wind and Fire. The songs represented a diversity of tempos (85 − 161 bpm, $M = 126$ bpm, $SD = 20$) and keys (9 represented). The list of songs can be found in Appendix Section A.1.

6.2 Procedure

Participants first completed a consent form and a demographic questionnaire. They were then introduced to our system and the adaptation parameters, and were given time to experiment and become familiar with the application controls. Subsequently, participants completed the study conditions, which involved designing adaptation parameters to integrate two notifications into each of the three songs for three different scenarios. The conditions were structured into three blocks, each corresponding to one of the three scenarios. Within each block, participants designed parameters for two notifications, one after the other. During the tasks, participants were asked to follow a think-aloud protocol, facilitated by the experimenter. Upon completion of all tasks, we conducted a semi-structured interview to gather their insights on (1) their approach and experience in designing parameters, (2) their impressions of the concept of adaptively embedding notifications in music, and (3) their suggestions for additional parameters.

6.3 Participants and apparatus

We invited six participants from a local university (6 male, age: $M = 26$ years, $SD = 1$). All participants had substantial musical experience ($M = 13$ years, $SD = 5$). One participant is a freelance performer, composer, and instrument manufacturer. Two participants had experience DJing and composing electronic music. One participant is a part-time jazz pianist. Two self-reported as music hobbyists. Participants received a $30 gift card as compensation.

The study was conducted using our MARingBA system implemented in a Unity 2021 Editor running on a MacBook Pro (macOS Ventura 13.4, 2.4 GHz Quad-Core Intel Core i5, stereo speakers with high dynamic range). All sessions were audio and video recorded. We recorded all final parameters for each condition.

6.4 Results

Throughout the study, participants recognized the potential benefits of blending notifications with music, especially in low-urgency and medium-urgency scenarios. They emphasized the importance of achieving a balanced integration, blending the notifications well while ensuring they remain perceivable amidst the music. During the exploration of various parameters, participants reported that beat matching, key matching, fade-in, and volume control for both notifications and music are the most prioritized factors for achieving the desired effect.

While timbre transfer, reverb, and low/high pass were also prioritized by multiple participants, they also expressed concern that these parameters may fail to generalize across different songs. While a certain instrument timbre blends well with the instrumentation of one song, it may create vastly different effects when integrated into another song, making it hard to control when designing for a certain level of urgency. Additionally, participants highlighted the significance of retaining the original timbre for notifications, particularly in high-urgency scenarios, as it was strongly associated with the familiarity of the ringtone.

The study demonstrated that the system provided sufficient coverage of parameters, allowing participants to fine-tune the integration of notifications based on different musical contexts and urgency levels. However, few participants expressed challenges in generalizing parameters across different songs or sections of the same song, where tempo, key, instrumentation, and volume can vary significantly. This revealed a potential need for more adaptive parameters that can respond to various musical contexts.

Another noteworthy advantage of the system was its added choice and customization options. Prior to using MARingBA, many participants mentioned that they often turned off notifications altogether for low-urgency scenarios to avoid disruptions. However, with the introduction of the blending notifications approach, content creators were

open to enabling notifications for low-urgency situations without the fear of being startled by disruptive sounds. We believe that this points to a good balance between staying informed and preserving their listening experience, leading to increased overall satisfaction with the notification system.

The average parameters of all experts for the different levels of urgency can be found in Table 1. We use those parameters for the notification in the second study.

7 EMPIRICAL STUDY

To further explore the benefits and limitations of adapting notifications to a background musical context, we conducted an empirical study to examine the effect of different notification adaptations on end-user task performance and experience. 12 participants experienced notifications adapted using the parameters elicited from our expert study while performing a typing task. Our study investigates the following research questions: (**RQ1**) To what extent do different adaptations modulate notification noticeability? (**RQ2**) To what extent do our notification adaptations affect a user's task performance? (**RQ3**) What aspects do users like and dislike about our approach to adapting notifications?

7.1 Design

We used a single-variable within-subject design with four ADAPTATION METHODS (*standard, low urgency, medium urgency, high urgency)*. Inspired by previous research on interrupts (e. g., [17, 18]), we adopted a dual-task paradigm where participants performed a PRIMARY TASK of typing while listening and responding to audio notifications manipulated with our ADAPTATION METHODS as a SECONDARY TASK. Participants experienced notifications adjusted using each ADAPTATION METHODS twice, each time for a different NOTIFICATION and SONG (i. e., 4 ADAPTATION METHOD × 2 NOTIFICATION-SONG pairs = 8 repetitions). NOTIFICATIONS and SONGS were randomly selected from the same pool (with 2 songs removed, one was shorter than 3 minutes, and the other had a section with very drastic tempo changes which may lead to unexpected adaptation results) as in the design study (Section 6). We counterbalanced the order of ADAPTATION METHODS across participants using a Latin Square.

7.1.1 Adaptation method. To generate the parameters for the *low urgency*, *medium urgency*, and *high urgency* ADAPTATION METHOD conditions, we used the mean of the parameter settings generated from our design study (i. e., from the six participants) for continuous values and the mode for categorical values. For the *standard* condition, we set the notification to mute the background music following conventional delivery approaches (section 3.2). All parameters can be found in Table 1.

7.1.2 Primary task. Participants were instructed to transcribe articles from Wikipedia as quickly and accurately as possible. The interface is shown in Figure 8. As visual feedback for the task, completed words are highlighted in green, typed letters of the current word are highlighted in yellow, and incorrectly typed characters turned the current letter red. Participants performed the typing task for three minutes in each condition.

7.1.3 Secondary task. While participants performed the typing task, they were asked to simultaneously monitor for audio notifications. Participants were instructed to click a button (see Figure 8, top right) when they noticed a notification as soon as possible. The button temporarily turned black to provide visual feedback for the response. In each condition, three notifications were delivered at randomized intervals at least 40 seconds apart, with the earliest and latest appearance at 0:10 minutes and 2:50 minutes, respectively.

Parameter	Low Urgency	Medium Urgency	High Urgency	Standard
Beat-match	True	True	True	False
Key-match	True	True	True	False
Panning Weight	22.71	25.72	35.47	0
Adaptive Volume Weight	10.71	12.25	13.92	0
Ringtone Volume	23.0	26.09	28.55	30
Timbre Transfer	Original	Original	Original	Original
Fade-In	True	False	False	False
Fade-In Speed	0.13	Not Used	Not Used	Not Used
Reverb Decay	10.43	8.1	3.94	0
Reverb Dry/Wet	476.18	351.45	141.36	0
Lowpass	17994.91	17994.91	16364.91	20000
Highpass	411.0	151.09	82.91	10
Replace Vocal	False	True	True	True
Replace Song	False	False	True	True
Song Volume Decrease	Not Used	-8.88	-15.9	-30
Schedule Mode	Next Four Bar	Next Four Bar	Next Beat	Not Used

Table 1. Parameter settings for each delivery method, derived from our notification design study (section 6).

We designed our study for a scenario where end users listen to their personal music collection. We therefore asked all users to rank the familiarity of the twelve songs used in the first expert study, and used songs that were familiar to them. We also made sure to cover a wide range of songs, and thus did not always select the most-familiar songs.

7.2 Procedure

After participants completed a consent form and a demographic questionnaire, they were introduced to the study tasks. Subsequently, they completed the study conditions, which were structured into two blocks. Each block consists of four repetitions of the dual-task paradigm for a randomly selected pair of song and notification. In each repetition, a different adaptation method was applied. Participants responded to questionnaires at the end of each condition. They

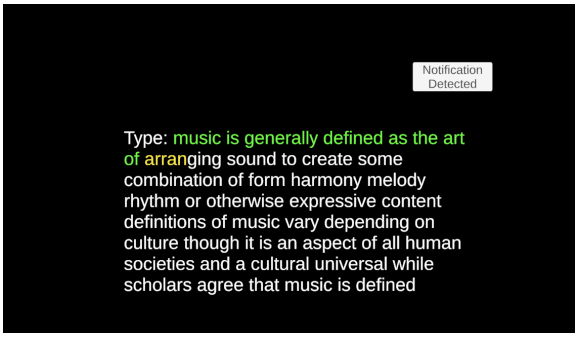

Fig. 8. Interface for the empirical study. Participants transcribed articles from Wikipedia. Correctly typed words are highlighted in green. The typed letters of the current word are highlighted in yellow. Upon noticing a notification, participants clicked the *notification detected* button on the top right.

also ranked the conditions by preference at the end of each block. After completing all conditions, participants reported on their overall experience with the different notification delivery mechanisms.

7.3 Participants and apparatus

We recruited twelve participants from a local university and convenience sampling (7 male, 5 female, age: $M = 24.67$ years, $SD = 3.75$). Participants listened to music daily ($M = 3$ hours, $SD = 1$) during activities like work ($N = 10$) and exercise ($N = 8$), as well as during their commute ($N = 10$). Participants received a $15 gift card as compensation.

We integrated the typing and notification response tasks into our MARingBA system, implemented in Unity 2021. The study was conducted on a MacBook Pro (macOS Ventura 13.4, 2.4 GHz Quad-Core Intel Core i5) with a pair of AKG K240 Studio over-ear headphones.

7.4 Measures

As dependent variables, we captured participants' primary and secondary task performance and subjective experience.

- **Primary task performance:** We measured *typing errors*, i. e., the number of incorrectly typed keys, and *resumption lag*, i. e., the time between notification detection and resuming to a regular typing speed.
- **Secondary task performance:** We measured *reaction time* as the elapsed time between the start of the notification and the participant's response, as well as the number of missed notifications.
- **Self-reported metrics:** At the end of each condition, participants reported their *confidence* and *immediacy* in detecting the presented notifications, and characterized the notifications in terms of their *noticeability* and *distraction*, all on a scale from 1 (low) to 7 (high). At the end of each block, participants provided a preference *ranking* of the adaptation methods.

7.5 Results

For effect analysis, ordinal data (questionnaire ratings and rankings) were analyzed using Friedman tests, and Wilcoxon signed-rank tests for post-hoc analysis when needed. Interval data was analyzed using a repeated measured ANOVA. In cases where the normality assumption was violated (Shapiro-Wilk test $p < .05$), we applied an Aligned Rank Transform (ART) prior to performing our analysis [48]. When needed, pairwise post-hoc tests (Bonferroni adjusted p-values) were performed. For each variable, the PARTICIPANT was considered as a random factor and the ADAPTATION METHOD as a within-subject factor. The statistical analysis was performed in IBM SPSS Statistics 29.

7.5.1 Primary task performance. We did not observe significant main effects on *typing errors* ($p = 0.452$) or *resumption lag* ($p = 0.861$). Participants made $M = 46.94$ typos, $SD = 21.18$, and exhibited a resumption lag of $M = 8.68\,s$, $SD = 1.90$. Individual conditions were within ~2% of the mean for typos, and ~4% of the mean for resumption lag.

7.5.2 Secondary task performance. Questionnaire responses are shown in Figure 9. Across participants, a total of 288 notifications were delivered during the experiment. We found a main effect of ADAPTATION METHOD on *reaction time* ($F_{3,33} = 11.612$, $p < .001$). On average, participants were significantly faster at responding to notifications in the *standard* ($M = 16.29\,s$, $SD = 3.69$, $p < 0.001$) and *high urgency* ($M = 20.29\,s$, $SD = 4.30$, $p < 0.012$) conditions compared to the *low urgency* condition ($M = 34.33\,s$, $SD = 2.314$). Participants missed three notifications in the *low urgency* condition, and one notification in the *medium urgency* condition. This highlights that all types of ADAPTATION METHODS are well suited to deliver notifications that are noticed, albeit later reaction time for lower-urgency ones.

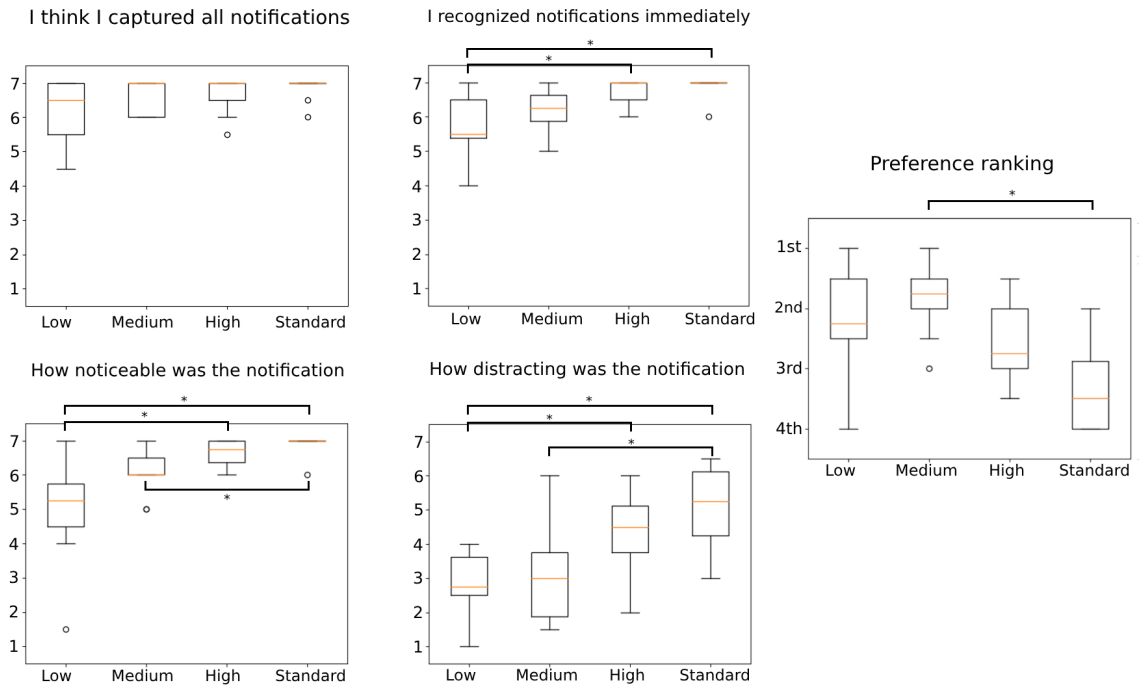

Fig. 9. Participant's subjective ratings across delivery methods on a scale from 1 (low) to 7 (high), ranking of preference from 1st (most preferred) to 4th (least preferred). Horizontal bars indicate statistically significant differences ($p < .05$) between conditions.

7.5.3 Self-reported metrics. We found a main effect of ADAPTATION METHOD on *confidence* ($\chi^2(3) = 14.234$, $p = 0.003$), *immediacy* ($\chi^2(3) = 24.786$, $p < 0.001$), *noticeability* ($\chi^2(3) = 24.677$, $p < 0.001$), *distraction* ($\chi^2(3) = 24.636$, $p < 0.001$), and *ranking* ($\chi^2(3) = 13.622$, $p = 0.003$).

Compared to the *standard* ADAPTATION METHOD, participants first regarded both the *low urgency* and *medium urgency* ADAPTATION METHODS as less *noticeable* (*low urgency*: $p = 0.03$, *medium urgency*: $p = 0.042$) and less *distracting* (*low urgency*: $p = 0.012$, *medium urgency*: $p = 0.042$). They also reported responding to the *standard* ADAPTATION METHOD more *immediately* than the *low urgency* ADAPTATION METHOD ($p = 0.03$). Compared to the *high urgency* ADAPTATION METHOD, participants regarded the *low urgency* ADAPTATION METHOD as less *noticeable* ($p = 0.03$) and less *distracting* ($p = 0.03$). They also reported responding to the *high urgency* ADAPTATION METHOD more immediately than the *low urgency* ADAPTATION METHOD ($p = 0.03$). Lastly, the reported preferring *medium urgency* ADAPTATION METHOD over the *standard* ADAPTATION METHOD ($p = 0.042$).

7.5.4 Discussion. Participants were generally highly successful in detecting notifications. While we believe that this can be attributed to the fact that they were aware that notifications would happen (i. e., cued detection), it nevertheless confirms that even notifications designed for low urgency are generally noticeable. Participants' recognition speed and subjective ratings on perceived noticeability largely align, with *standard* and *high urgency* being rated more noticeable than the other two conditions. The ratings on distraction are directly related to noticeability, with *standard* and *high urgency* being perceived as more distracting. Qualitative comments, however, also point in a different direction, where low noticeability is also perceived as distracting, as noted by one participant "I liked the way that the earlier styles [low

urgency] were woven into the music rather than causing the music to stop or grow very faint, but depending on the song this might be more annoying and could be rather hard to detect." (P9) Participants generally preferred a balance between noticeable and distracting, seemingly best fulfilled by the medium-urgency condition, as reflected in comments such as "While the more loud ones definitely caught my attention, I think that the more in-between notification sounds were ideal." (P5)

Participants were also able to identify, or at least appreciate, certain manipulations MARingBA performed. One participant reflected on how they enjoyed volume adjustments including track-specific volume attenuation and fading: "I like it when the notifications come in, the music volume will be slightly reduced and the notification volume gradually increases. I don't like the places where the notifications just kick in without easing." (P7) Finally, even though some participants could not clearly identify certain features of our approach, they perceived beat and key matching as desirable, as noted by one participant: "I do not know if it was an accident but one of the notification sound came up exactly on the same beat of the Happy song, and I felt that it was actually nice for the notification to "enter" the song in the same tempo." (P12) We believe that these results indicate that our approach struck a good balance between noticeability and distraction, and that participants generally appreciate the music-adaptive notification, particularly compared to the standard delivery approach.

8 SCENARIOS OF USAGE

In the following, we provide examples where we envision that music-adaptive notifications created with MARingBA provide a superior experience compared to standard delivery methods. A key factor is that with our system, audio notifications can be delivered to match varying levels of urgency.

- **Scenario 1:** A user is jogging through the city, listening to the song "One" by U2. The calendar on their mobile phone sends an audio notification reminding them that they need to meet a friend in one hour. MARingBA modifies the timbre of the notification to sound like a piano to match the style, and matches the beat of the song to integrate it better. Additionally, it plays it in a lower volume, since the notification does not require immediate attention.
- **Scenario 2:** A user is deeply engrossed in their work, accompanied by a calming lofi hip hop playlist tailored for productive studying. Typically, they'd mute all notifications to maintain their focus. However, since the user recently applied to multiple jobs, they might receive important text messages for interviews that require a timely response, a medium-urgency scenario. Nevertheless, users should have sufficient time to finish their current subtask before attending to the message. MARingBA blends to incoming audio notification of a text message seamlessly into the rhythmic drum beat of the lofi tunes. Slowly increasing in volume, the notification naturally captures the user's attention. They adeptly address the task without disrupting their flow, underscoring how technology can facilitate a harmonious balance between productivity and responsiveness.
- **Scenario 3:** At a lively party, the user and their friends groove to a continuous dance music mix. Amidst the celebration, the user has an important task to attend to: taking the pizza out of the oven before it burns. To ensure they don't forget, they've set a timer. Instead of an abrupt alarm, the timer seamlessly integrates a custom notification sound, that is only recognized by the user, into the dance mix. The user catches the subtle reminder, grabs the pizza, and rejoins the party, without disrupting the rhythm of the dance floor.

9 DISCUSSION

We contribute a novel music-adaptive approach to delivering audio notifications such as ringtones. Our work explores the design space of possible audio manipulations such as beat matching, key matching, or timbre adjustments. We integrate those manipulations into MARingBA, a novel system that enables content creators to design adaptive audio notifications. We explore our system in a design study with experts, who design a variety of ringtones for different songs. Insights from the study indicate that MARingBA enables them to explore the parameter space efficiently. We use the parameters obtained in the design study in an evaluation with end users. Results indicate that our music-adaptive audio notifications designed for different levels of urgency provide users with noticeable signals with varying levels of reaction times, and are preferred over a standard delivery baseline. We believe that music-adaptive notifications are a feasible complement, or even replacement, for current notification delivery methods. There exists, however, still unexplored areas of potential challenges and opportunities that we hope to explore in the future.

9.1 Personalization

The settings we tested, as well as the parameters of the audio notifications that were created by the experts in the design study, worked well across the different songs and led to desirable task performance. Low-urgency notifications were perceived later than high-urgency ones, and perceived as less distracting. In the qualitative results, however, we saw that a number of participants preferred the low-urgency notifications, whereas others wanted more salient signals like the medium-urgency settings. This hints at opportunities for personalization. Similar to current ringtones in phones, we believe that future versions of our approach should allow users to personalize their settings, and give end-users agency in the design of the notifications. We are eager to explore the granularity of such personalization, from a single value parameter for "strength", to giving users control over individual parameters such as timbre or key matching.

9.2 Generalizability

We see the implementation of MARingBA as an initial exploration of the very large parameter space. Music preferences across content creators and end users vary, and we do not believe that a one-size-fits-all solution exists. Our tool for content creators allows for a certain flexibility in the design, but is limited by what parameters are currently implemented. We hope to perform more longitudinal studies with both content creators and end users in the future. This involves a more systematic exploration of different audio notifications and notification types (e. g., different lengths, voice notifications, etc.), and exploration of different musical genres.

In our second evaluation, we ensured that users were reasonably familiar with the songs they listened to, assuming that participants typically listen to music they like and know. Few participants, however, were presented with less familiar songs (one participant with a completely unfamiliar song), and their performance did not change. We therefore believe that our approach generalizes beyond the songs we used in our evaluation.

9.3 Multi-modal notifications

Our current approach is focused on audio notifications. Current interactive systems, however, deliver content and information through a wide range of modalities, from visual such as desktop or Mixed Reality settings, to haptic notifications through vibrations on a smartphone, or smell and taste. We plan to pair music-adaptive audio adaptations with other modalities in the future to investigate how well those can integrated into a multi-modal delivery mechanism. Additionally, we hope to explore applying our adaptive approach to other modalities. One could easily imagine an

approach where the vibration of a haptic notification for a phone call is synchronized to the music that a user is currently listening to; or that visual notifications in Mixed Reality appear in a style that matches the musical genre. We are excited to explore those combinations in the future, and find out what modalities are best suited for musical adaptations to be less disruptive, or might even enhance the music listening experience for users.

9.4 In-the-wild studies

We currently evaluate our approach with end users in a highly controlled environment, where we control the space, task, and what music participants were listening to.

We hope to expand our evaluation to more users and more context in the future. Other contexts such as tasks (e. g., running, shopping) and spaces (e. g., indoor, outdoor) will inevitably influence participants' ability to detect audio notifications such as ringtones. Additionally, we believe that particularly the question of notification scheduling is interesting for further investigation. For example, delaying notifications to opportune moments in later sections of the song, or even the next song, might be desirable in scenarios of focused work to minimize switching costs. In other scenarios, such as phone calls, the delivery cannot be delayed drastically, since the caller might hang up. We hope to explore this aspect in the future. Finally, we hope to expand our music-adaptive approach with physiological sensing [20] in the future to balance noticeability, urgency, and distraction. We believe that our music-adaptive content delivery is a first step towards context-aware delivery of audio notifications that are noticeable, not distracting, and ultimately beneficial for end users.

10 CONCLUSION

We contribute a novel approach to creating music-adaptive audio notifications by blending them into songs, leveraging a range of parameters such as beat, key, or timbre. An expert study confirms that our approach is valuable for designers of notifications. An evaluation confirms that our parametrization of audio notifications directly influences noticeability and perceived distraction, and is preferred by end users. We believe that music-adaptive audio notifications have the potential to complement or replace current standard delivery methods such as volume fading, and provide users with timely access to information without being disruptive. We started an exploration of a very large parameter space, that can be used for a range of application scenarios. We believe that our work lays the groundwork for future context-aware interactive systems that adapt audio notifications and non-visual digital content based on users' current music, surroundings, and tasks.

REFERENCES

[1] Piotr D. Adamczyk and Brian P. Bailey. 2004. If Not Now, When? The Effects of Interruption at Different Moments within Task Execution. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vienna, Austria) *(CHI '04)*. Association for Computing Machinery, New York, NY, USA, 271–278. https://doi.org/10.1145/985692.985727

[2] Piotr D Adamczyk and Brian P Bailey. 2004. If not now, when? The effects of interruption at different moments within task execution. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 271–278.

[3] Ishwarya Ananthabhotla and Joseph A. Paradiso. 2018. SoundSignaling: Realtime, Stylistic Modification of a Personal Music Corpus for Information Delivery. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 4, Article 154 (dec 2018), 23 pages. https://doi.org/10.1145/3287032

[4] E. Arroyo, T. Selker, and A. Stouffs. 2002. Interruptions as multimodal outputs: which are the less disruptive?. In *Proceedings. Fourth IEEE International Conference on Multimodal Interfaces*. 479–482. https://doi.org/10.1109/ICMI.2002.1167043

[5] Brian P. Bailey and Joseph A. Konstan. 2006. On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state. *Computers in Human Behavior* 22, 4 (2006), 685–708. https://doi.org/10.1016/j.chb.2005.12.009 Attention aware systems.

[6] Brian P Bailey, Joseph A Konstan, and John V Carlis. 2001. The Effects of Interruptions on Task Performance, Annoyance, and Anxiety in the User Interface.. In *Interact*, Vol. 1. 593–601.

[7] Luke Barrington, Michael J. Lyons, Dominique Diegmann, and Shinji Abe. 2006. Ambient Display Using Musical Effects. In *Proceedings of the 11th International Conference on Intelligent User Interfaces* (Sydney, Australia) *(IUI '06)*. Association for Computing Machinery, New York, NY, USA, 372–374. https://doi.org/10.1145/1111449.1111541

[8] Alfred Blatter. 2016. *Revisiting music theory: basic principles*. Taylor & Francis.

[9] Meera M. Blattner, Denise A. Sumikawa, and Robert M. Greenberg. 1989. Earcons and Icons: Their Structure and Common Design Principles. *Human–Computer Interaction* 4, 1 (1989), 11–44. https://doi.org/10.1207/s15327051hci0401_1

[10] Sebastian Böck, Filip Korzeniowski, Jan Schlüter, Florian Krebs, and Gerhard Widmer. 2016. Madmom: A New Python Audio and Music Signal Processing Library. In *Proceedings of the 24th ACM International Conference on Multimedia* (Amsterdam, The Netherlands) *(MM '16)*. Association for Computing Machinery, New York, NY, USA, 1174–1178. https://doi.org/10.1145/2964284.2973795

[11] Deborah A. Boehm-Davis and Roger Remington. 2009. Reducing the disruptive effects of interruption: A cognitive framework for analysing the costs and benefits of intervention strategies. *Accident Analysis & Prevention* 41, 5 (2009), 1124–1129. https://doi.org/10.1016/j.aap.2009.06.029

[12] Dmitry Bogdanov, Nicolas Wack, Emilia Gómez, Sankalp Gulati, Perfecto Herrera, Oscar Mayor, Gerard Roma, Justin Salamon, José Zapata, and Xavier Serra. 2013. ESSENTIA: An Open-Source Library for Sound and Music Analysis. In *Proceedings of the 21st ACM International Conference on Multimedia* (Barcelona, Spain) *(MM '13)*. Association for Computing Machinery, New York, NY, USA, 855–858. https://doi.org/10.1145/2502081.2502229

[13] Stephen Brewster. 2007. Nonspeech auditory output. In *The human-computer interaction handbook*. CRC Press, 273–290.

[14] Andreas Butz and Ralf Jung. 2005. Seamless User Notification in Ambient Soundscapes. In *Proceedings of the 10th International Conference on Intelligent User Interfaces* (San Diego, California, USA) *(IUI '05)*. Association for Computing Machinery, New York, NY, USA, 320–322. https://doi.org/10.1145/1040830.1040914

[15] Nick Collins. 2010. *Introduction to computer music*. John Wiley & Sons.

[16] Fulvio Corno, Luigi De Russis, and Teodoro Montanaro. 2017. XDN: Cross-Device Framework for Custom Notifications Management. In *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (Lisbon, Portugal) *(EICS '17)*. Association for Computing Machinery, New York, NY, USA, 57–62. https://doi.org/10.1145/3102113.3102127

[17] Edward B. Cutrell, Mary Czerwinski, and Eric Horvitz. 2000. Effects of Instant Messaging Interruptions on Computing Tasks. In *CHI '00 Extended Abstracts on Human Factors in Computing Systems* (The Hague, The Netherlands) *(CHI EA '00)*. Association for Computing Machinery, New York, NY, USA, 99–100. https://doi.org/10.1145/633292.633351

[18] Mary Czerwinski, Edward Cutrell, and Eric Horvitz. 2000. Instant messaging and interruption: Influence of task type on performance. In *OZCHI 2000 conference proceedings*, Vol. 356. 361–367.

[19] Matthew E. P. Davies, Philippe Hamel, Kazuyoshi Yoshii, and Masataka Goto. 2014. AutoMashUpper: Automatic Creation of Multi-Song Music Mashups. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22, 12 (2014), 1726–1737. https://doi.org/10.1109/TASLP.2014.2347135

[20] Pascal E Fortin, Elisabeth Sulmont, and Jeremy Cooperstock. 2019. Detecting perception of smartphone notifications using skin conductance responses. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–9. https://doi.org/10.1145/3290605.3300420

[21] Stavros Garzonis, Simon Jones, Tim Jay, and Eamonn O'Neill. 2009. Auditory icon and earcon mobile service notifications: intuitiveness, learnability, memorability and preference. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1513–1522.

[22] William W. Gaver. 1993. Synthesizing Auditory Icons. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* (Amsterdam, The Netherlands) *(CHI '93)*. Association for Computing Machinery, New York, NY, USA, 228–235. https://doi.org/10.1145/169059.169184

[23] Sarthak Ghosh, Lauren Winston, Nishant Panchal, Philippe Kimura-Thollander, Jeff Hotnog, Douglas Cheong, Gabriel Reyes, and Gregory D. Abowd. 2018. NotifiVR: Exploring Interruptions and Notifications in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics* 24, 4 (2018), 1447–1456. https://doi.org/10.1109/TVCG.2018.2793698

[24] Jennifer Gluck, Andrea Bunt, and Joanna McGrenere. 2007. Matching Attentional Draw with Utility in Interruption. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 41–50. https://doi.org/10.1145/1240624.1240631

[25] Robert Graham. 1999. Use of auditory icons as emergency warnings: evaluation within a vehicle collision avoidance application. *Ergonomics* 42, 9 (1999), 1233–1248.

[26] Romain Hennequin, Anis Khlif, Felix Voituret, and Manuel Moussallam. 2020. Spleeter: a fast and efficient music source separation tool with pre-trained models. *Journal of Open Source Software* 5, 50 (2020), 2154.

[27] Edward Cutrell Mary Czerwinski Eric Horvitz. 2001. Notification, disruption, and memory: Effects of messaging interruptions on memory and performance. In *Human-Computer Interaction: INTERACT*, Vol. 1. 263.

[28] Scott Hudson, James Fogarty, Christopher Atkeson, Daniel Avrahami, Jodi Forlizzi, Sara Kiesler, Johnny Lee, and Jie Yang. 2003. Predicting Human Interruptibility with Sensors: A Wizard of Oz Feasibility Study. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) *(CHI '03)*. ACM, New York, NY, USA, 257–264. https://doi.org/10.1145/642611.642657

[29] Shamsi T. Iqbal and Brian P. Bailey. 2006. Leveraging Characteristics of Task Structure to Predict the Cost of Interruption. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Québec, Canada) *(CHI '06)*. Association for Computing Machinery, New York, NY, USA, 741–750. https://doi.org/10.1145/1124772.1124882

[30] Shamsi T. Iqbal and Brian P. Bailey. 2011. Oasis: A Framework for Linking Notification Delivery to the Perceptual Structure of Goal-Directed Tasks. *ACM Trans. Comput.-Hum. Interact.* 17, 4, Article 15 (dec 2011), 28 pages. https://doi.org/10.1145/1879831.1879833

[31] Hiromi Ishizaki, Keiichiro Hoashi, and Yasuhiro Takishima. 2009. Full-Automatic DJ Mixing System with Optimal Tempo Adjustment based on Measurement Function of User Discomfort. In *International Society for Music Information Retrieval Conference*. https://api.semanticscholar.org/CorpusID:6179832

[32] Ralf Jung. 2008. Ambience for auditory displays: Embedded musical instruments as peripheral audio cues. In *Proc. ICAD*.

[33] Mohamed Kari, Tobias Grosse-Puppendahl, Alexander Jagaciak, David Bethge, Reinhard Schütte, and Christian Holz. 2021. SoundsRide: Affordance-synchronized music mixing for in-car audio augmented reality. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event USA). ACM, New York, NY, USA. https://doi.org/10.1145/3472749.3474739

[34] Thomas Kubitza, Alexandra Voit, Dominik Weber, and Albrecht Schmidt. 2016. An IoT Infrastructure for Ubiquitous Notifications in Intelligent Living Environments. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct* (Heidelberg, Germany) *(UbiComp '16)*. Association for Computing Machinery, New York, NY, USA, 1536–1541. https://doi.org/10.1145/2968219.2968545

[35] Uichin Lee, Joonwon Lee, Minsam Ko, Changhun Lee, Yuhwan Kim, Subin Yang, Koji Yatani, Gahgene Gweon, Kyong-Mee Chung, and Junehwa Song. 2014. Hooked on Smartphones: An Exploratory Study on Smartphone Overuse among College Students. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 2327–2336. https://doi.org/10.1145/2556288.2557366

[36] Paul MC Lemmens, Myra P Bussemakers, and Abraham De Haan. 2001. Effects of auditory icons and earcons on visual categorization: the bigger picture. In *Proceedings of the International Conference on Auditory Display*. 117–125.

[37] Aristides Mairena, Carl Gutwin, and Andy Cockburn. 2019. Peripheral Notifications in Large Displays: Effects of Feature Combination and Task Interference. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300870

[38] Tara Matthews, Anind K. Dey, Jennifer Mankoff, Scott Carter, and Tye Rattenbury. 2004. A Toolkit for Managing User Attention in Peripheral Displays. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology* (Santa Fe, NM, USA) *(UIST '04)*. Association for Computing Machinery, New York, NY, USA, 247–256. https://doi.org/10.1145/1029632.1029676

[39] D. Scott McCrickard and C. M. Chewar. 2003. Attuning Notification Design to User Goals and Attention Costs. *Commun. ACM* 46, 3 (mar 2003), 67–72. https://doi.org/10.1145/636772.636800

[40] Brian McFee, Colin Raffel, Dawen Liang, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and music signal analysis in python.

[41] Abhinav Mehrotra, Veljko Pejovic, Jo Vermeulen, Robert Hendley, and Mirco Musolesi. 2016. My phone and me: understanding people's receptivity to mobile notifications. In *Proceedings of the 2016 CHI conference on human factors in computing systems*. 1021–1032.

[42] Philipp Müller, Sander Staal, Mihai Bâce, and Andreas Bulling. 2022. Designing for Noticeability: Understanding the Impact of Visual Importance on Desktop Notifications. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 472, 13 pages. https://doi.org/10.1145/3491102.3501954

[43] Richard W. Obermayer and William A. Nugent. 2000. Human-computer interaction for alert warning and attention allocation systems of the multimodal watchstation. In *Integrated Command Environments*, Patricia Hamburger (Ed.), Vol. 4126. International Society for Optics and Photonics, SPIE, 14 – 22. https://doi.org/10.1117/12.407536

[44] Martin Pielot, Karen Church, and Rodrigo de Oliveira. 2014. An In-Situ Study of Mobile Phone Notifications. In *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices & Services* (Toronto, ON, Canada) *(MobileHCI '14)*. Association for Computing Machinery, New York, NY, USA, 233–242. https://doi.org/10.1145/2628363.2628364

[45] Marco A Martinez Ramirez, Weihsiang Liao, Chihiro Nagashima, Giorgio Fabbro, Stefan Uhlich, and Yuki Mitsufuji. 2022. Automatic music mixing with deep learning and out- of-domain data. In *Proceedings of the 23rd International Society for Music Information Retrieval Conference*. ISMIR, Bengaluru, India, 411–418. https://doi.org/10.5281/zenodo.7316688

[46] Alexandra Voit, Tonja Machulla, Dominik Weber, Valentin Schwind, Stefan Schneegass, and Niels Henze. 2016. Exploring Notifications in Smart Home Environments. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct* (Florence, Italy) *(MobileHCI '16)*. Association for Computing Machinery, New York, NY, USA, 942–947. https://doi.org/10.1145/2957265.2962661

[47] Dominik Weber, Alireza Sahami Shirazi, and Niels Henze. 2015. Towards Smart Notifications Using Research in the Large. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct* (Copenhagen, Denmark) *(MobileHCI '15)*. Association for Computing Machinery, New York, NY, USA, 1117–1122. https://doi.org/10.1145/2786567.2794334

[48] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 143–146. https://doi.org/10.1145/1978942.1978963

[49] Jing Yang, Tristan Cinquin, and Gábor Sörös. 2021. Unsupervised Musical Timbre Transfer for Notification Sounds. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 3735–3739. https://doi.org/10.1109/ICASSP39728.2021.9414760

[50] Jing Yang and Andreas Roth. 2021. Musical Features Modification for Less Intrusive Delivery of Popular Notification Sounds. *Proceedings of the 26th International Conference on Auditory Display (ICAD 2021)* (2021). https://api.semanticscholar.org/CorpusID:236204585

A APPENDIX

A.1 List of songs and notifications

List of notifications used in both studies:

(1) Discord
(2) Google hangouts
(3) iPhone default
(4) Skype
(5) iPhone classic (marimba)
(6) Line (not used in empirical study)

List of songs used in both studies (song, artist):

(1) Blinding lights, The Weeknd
(2) Counting stars, OneRepublic (not used in empirical study)
(3) Happy, Pharrell Williams
(4) Never gonna give you up, Rick Astley
(5) Nothin on you, Bruno Mars
(6) September, Earth, wind and fire
(7) Somebody that I used to know, Gotye
(8) Toxic, Britney Spears
(9) What is love, Haddaway
(10) Mr Brightside, The Killers
(11) Aint no mountain high enough, Marvin Gaye and Tammi Terrell (not used in empirical study)
(12) Hotline bling, Drake