Towards Music-Aware Virtual Assistants

Alexander Wang
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA

David Lindlbauer
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA

Chris Donahue
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA

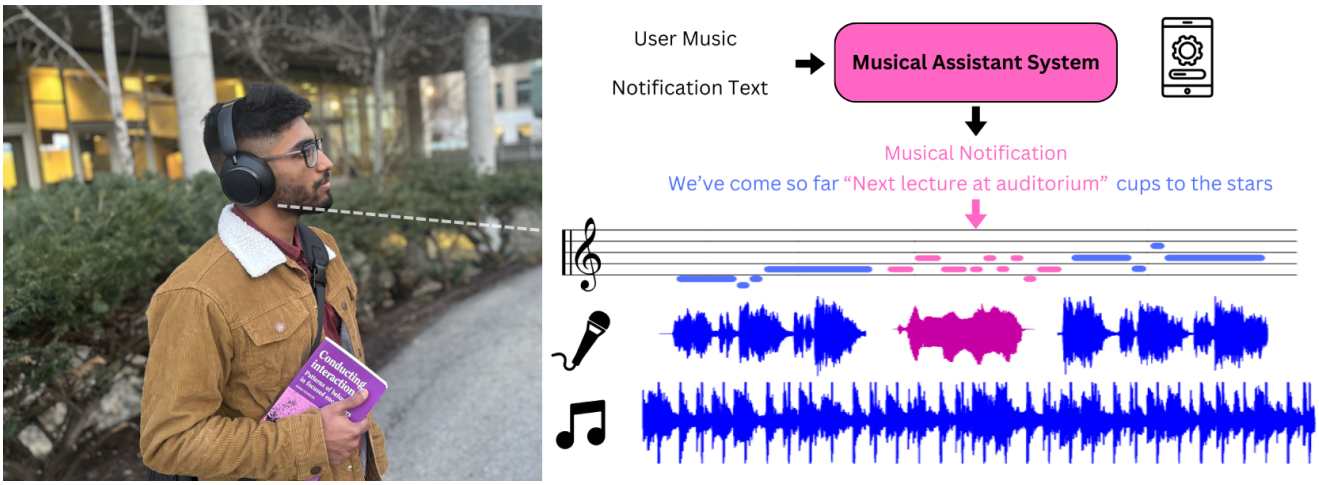

Figure 1: We propose music-aware virtual assistants, ones that more seamlessly integrate spoken notifications with a user's music. Given a user's music and notification text, our system inserts a musical notification by (1) synthesizing that text as a singing voice with a generated melody that agrees with both the surrounding musical context and the text, and (2) temporarily replacing the original vocals with the musical notification.

ABSTRACT

We propose a system for modifying spoken notifications in a manner that is sensitive to the music a user is listening to. Spoken notifications provide convenient access to rich information without the need for a screen. Virtual assistants see prevalent use in hands-free settings such as driving or exercising, activities where users also regularly enjoy listening to music. In such settings, virtual assistants will temporarily mute a user's music to improve intelligibility. However, users may perceive these interruptions as intrusive, negatively impacting their music-listening experience. To address this challenge, we propose the concept of music-aware virtual assistants, where speech notifications are modified to resemble a voice singing in harmony with the user's music. We contribute a system that processes user music and notification text to produce a blended mix, replacing original song lyrics with the notification content. In a user study comparing musical assistants to standard virtual assistants, participants expressed that musical assistants fit better with music, reduced intrusiveness, and provided a more delightful listening experience overall.

UIST '24, October 13–16, 2024, Pittsburgh, PA, USA

ACM ISBN 979-8-4007-0628-8/24/10
https://doi.org/10.1145/3654777.3676416

CCS CONCEPTS

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

KEYWORDS

Audio, Music, Virtual Assistants, Notification, Interruptions, Speech, Machine Learning

ACM Reference Format:
Alexander Wang, David Lindlbauer, and Chris Donahue. 2024. Towards Music-Aware Virtual Assistants. In *The 37th Annual ACM Symposium on User Interface Software and Technology (UIST '24), October 13–16, 2024, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3654777.3676416

1 INTRODUCTION

Virtual voice assistants, such as Siri and Alexa, are ubiquitous features of smart devices including phones, laptops, and speakers. Given a *text notification* from an application, these systems use text-to-speech (TTS) to dictate a *spoken notification* to inform users of new information. Virtual assistants are especially beneficial in scenarios where users are engaged in other tasks such as walking, exercising, shopping, driving, or browsing. While driving, for example, users can keep their eyes on the road while receiving timely audio navigation instructions.

Incidentally, these routine scenarios are also typical moments where users listen to music [40]. Listening to music while receiving spoken notifications from virtual assistants, however, can be

perceived as intrusive. Typically, music and voice instructions are combined by temporarily decreasing the volume of the music and overlaying the speech. As shown by previous research on integrating ringtones into music [52, 54, 55], users perceive temporary musical interruptions as distracting and detrimental to the music-listening experience. Audio notifications might mask a user's favorite part of a song, or unimportant notifications unnecessarily take attention away from a user's main task.

Previous research attempted to solve this by integrating auditory notifications such as ringtones into users' music through techniques such as timbre transfer [54, 55] and harmonic mixing [52]. While those techniques improved user experience by decreasing the disruptiveness of notifications, they were only designed to work with ringtones. Ringtones are short musical compositions tailored to notify users of new activity but do not convey the details of that activity. In our work, we instead focus on spoken notifications from virtual assistants such as navigation instructions, announcements, or messages. Spoken notifications provide users with more information than ringtones, but are more challenging to integrate as they are not already musical.

We propose the novel concept of *music-aware virtual assistants* or *musical assistants* for short. Our approach *blends spoken notifications from virtual assistants into music.* Instead of muting a user's music and overlaying a spoken notification, we modify the spoken notifications so that they resemble someone "singing" them in harmony with the song a user is currently listening to.

To integrate voice messages into music, we contribute a system that (1) generates a new musical melody consistent with a user's music and the text notification, (2) modifies a spoken notification to match the new melody, and (3) integrates the resulting *musical notification* into the original song (Figure 1). A significant challenge in developing such systems is ensuring the intelligibility of the musical notifications. Specifically, we identify two key factors that affect the intelligibility of musical notifications: (1) the compatibility of the melodic rhythm and the natural spoken rhythm (*prosody*) of the text transcripts, and (2) the performance of singing voice synthesis (SVS) systems. To improve rhythmic compatibility, we first estimate the natural spoken rhythm of the text using TTS and then generate a new melody that is close to this rhythm but also compatible with the surrounding musical context. We also found that state-of-the-art SVS systems often produce unintelligible output, even given pairs of melody and text with high rhythmic compatibility (such as the original melody and lyrics). Accordingly, we propose to modify the output of TTS systems to conform to the generated melody using signal processing, sacrificing the naturalness of SVS systems in favor of intelligibility, which we argue is more critical for our application. We explore speech recognition as a proxy for human intelligibility, showing that our TTS-based system achieves higher intelligibility than one based on SVS.

Our goal is to improve users' music listening experience by making notifications musically aware, thus reducing intrusiveness, improving music fit, and making the experience delightful. Our approach complements other modes of notification presentation, as opposed to replacing them. Current types of notification presentations work well for many scenarios, especially high-urgency ones where immediate attention is needed. Our approach targets scenarios when low to medium-urgency messages are delivered in casual listening situations, such as receiving a reminder during an exercise session; or receiving a meeting invitation while going for a walk. During these tasks, our approach provides an unobtrusive and lighthearted alternative to turning notifications off.

Existing pitch correction tools such as Autotune[48] can also make non-musical sounds feel musical by adjusting the pitch to the closest musical pitch. However, our approach generates a new melody based on the constraints of both user music context (tempo, harmony) and text context (prosody, syllables), inpainting a new melody that stylistically fits the current song and the message, resulting in better musical integration and better intelligibility.

In a user study comparing our music-aware virtual assistant to the standard practice of combining spoken notifications and volume reductions, end users rated our approach as being a better fit with music, less intrusive, and more delightful. Participants expressed that they value clarity, music continuity, and context-sensitivity in scenarios where spoken notifications coexist with music-listening, and generally endorsed music being altered for important notifications. While participants were able to transcribe notifications in both settings with comparable levels of accuracy (suggesting high intelligibility of our method), participants reported that the seamless quality of our method required more cognitive effort from the user. This falls in line with prior research on musically-integrated ringtones [52], where the most subtle settings required more cognitive effort and is therefore less preferred than a setting that struck a balance in saliency and integration. Despite concerns about the naturalness of the singing, participants saw the strong potential in our novel approach, which is the first that aims at providing a complement to conventional spoken voice messages towards well-integrated, melodic ones. Especially in instances where the user is engaged in activities without a digital screen, the musical assistant system can subtly provide timely information without substantially disrupting the music-listening experience, offering a delightful alternative to constant interruptions or silencing notifications. Examples of our approach can be found at *https://augmented-perception.org/publications/2024-singing-assistants.html*.

In summary, we make the following contributions:

- A system that automatically synthesizes and integrates intelligible musical notifications into pop songs.
- A novel method for generating new melodies in the middle of pop songs, with precise rhythmic control.
- Results and insights from a user study (n = 12) with end users, showing that our approach is perceived to be less intrusive, more musically fit, and more delightful.

2 RELATED WORK

Our work builds on prior research in auditory displays, speech and singing voice intelligibility, and music generation.

2.1 Virtual assistants

Virtual assistants (or personal assistants, voice-enabled assistants, etc. [50]) enable natural, conversation-like, interactions with users by leveraging different components such as speech recognition, task execution, and synthesizing a speech response output. They support users in a variety of tasks, such as while cooking, web search, communication, and controlling IoT devices (e. g., turning

lights on and off) [1]. They provide users with on-demand messages, information and notifications, and are particularly useful for enabling hands-free interactions [11]. In our work, we concentrate on a fundamental aspect of virtual assistants: synthesizing speech outputs to communicate with users. Our goal is to enable virtual assistants to present messages in a less obtrusive manner by blending them with background music users are currently listening to.

2.2 Music and auditory displays

Auditory displays, such as auditory icons and earcons [5], communicate information through sound. Typical examples include ringtones or short cues to indicate the availability of new messages. They have been used to improve the effectiveness of user interfaces, e. g., by combining them with graphical interfaces [23, 39]. One goal of auditory displays is to present information unobtrusively to users, for example by integrating them with the music users currently listen to. Jung and Butz [7, 30], for example, explored pre-composing soundscapes where certain musical elements are optional and will not strongly affect the music listening experience when removed, enabling these optional music snippets to be included as an indicator of notification. Ananthabhotla and Paradiso [2] embedded notifications by allowing users to listen to their own music library, and substituting auditory notifications with arbitrary effects to the original audio. They rely on the user's familiarity with the original music to be effective. Yang et al. [54, 55] apply timbre transfer on popular notification sounds to make them less intrusive. Wang et al. [52] utilized techniques from music information retrieval to modify ringtone notifications for automated harmonic blends. All those works focus on short ringtones, whereas we focus on voice messages.

Other work adjusts the music users listen to, for example for navigation. Soundsride adapts the user's music according to the affordances of statuses on the road (e. g., synchronizing the start of a musical section with the exit of a tunnel) [31]. Navigatone separates the individual elements of a music composition and spatializes some components to seamlessly guide the user [25].

Despite the widespread and concurrent usage of virtual assistants and music streaming, to the best of our knowledge, we are the first to integrate speech-based auditory displays into music.

2.3 Speech and singing voice synthesis

Recent advancements in machine learning have enabled TTS systems that can synthesize speech with high fidelity, intelligibility, and speaker diversity [19, 32, 34, 41, 42, 45, 53]. Similar methods have been applied to singing voice synthesis (SVS) [35, 36], though researchers have noted a scarcity of training data as an obstacle to intelligible SVS [13]. Singing voice *conversion* (SVC) systems are increasingly popular in music production [22] and can convert one singing voice to another with high intelligibility [33, 47], but they require human singing as input and accordingly are not practical for musical notifications. Even human singing can be hard to understand [10, 21], posing an obstacle to our setting where intelligibility is of critical importance. To synthesize singing with higher intelligibility than existing SVS systems, our system modifies outputs from TTS systems to sound more musical (at the cost of naturalness).

2.4 Prosody and intelligibility

Previous research has found that intelligibility is generally imperfect for sung lyrics and varies across genres. Listener transcription accuracy can be as low as 48 percent for classical music and an average of 72 percent for all genres explored [10]. Johnson et al. [29] confirmed several factors of music composition that improve intelligibility, such as matching the rhythm of music composition with the rhythm of speech prosody, only assigning one musical note to a syllable as opposed to multiple (melismatic singing), and using more commonly used words. Similarly, Collister and Huron compared sung and spoken words and showed that spoken words are better understood by listeners [9].

One major factor for intelligibility in speech is prosody, i. e., the acoustic parameters of speech that shape the sound qualities beyond the textual context. For instance, it is hard to understand the speech of someone who speaks monotonously and stretches syllables to be the same duration. While the exact definition of prosody may differ between fields, Cutler and Ladd provide a concrete definition of prosody as "those phenomena that involve the acoustic parameters of pitch, duration, and intensity [12]." Research on the speech-to-song illusion shows that these acoustic features in speech have equivalents in music composition and can be interpreted as musical when being listened to repeatedly [15].

Building on research in music intelligibility, we aim to make the outputs of musical voice assistants intelligible through the guidelines proposed by previous researchers, including assigning messages to a melody that matches the prosody of the original speech. To achieve this, we build on work done in symbolic music generation to automatically compose melodies that are suitable for the input text.

2.5 Music generation

A number of recent works explore unconditional music generation by modeling symbolic music representations with autoregressive language models (LMs) [17, 28, 43, 46]. Our goal is not unconditional generation, but rather to generate melodies conditioned on input musical context (chords, beats) alongside target "lyrics" (the text content of the musical notification). Yu et al. [56] and Choi et al. [8] explore melody generation based on input lyrics and chords, respectively, but neither method can support both constraints simultaneously. Thickstun et al. [49] propose an LM-based symbolic music generation model that can both generate unconditionally and also *accompany*, i. e., generate music for one instrument to be played simultaneously with music from another. Here we extend this approach to be capable of generating melody conditioned on both chords from the music that the user is listening to, as well as the natural prosody of the target text.

3 MUSIC-AWARE VIRTUAL ASSISTANTS

We contribute a system that takes in notification text and user music as input and outputs musical notifications, illustrated in Figure 2. This output can then be integrated into user music by replacing any existing vocals for a blended delivery of information. We contribute two core components that improve the outputs of musical assistant systems, which enable our overall system. Specifically, we contribute (1) a novel method to generate new melodies by adjusting a

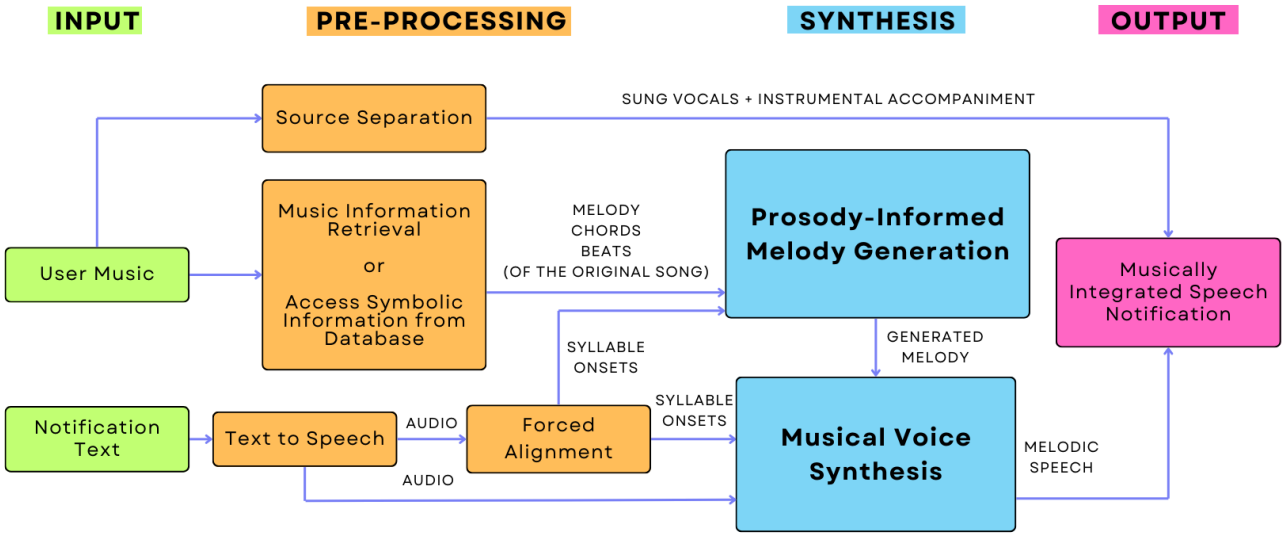

Figure 2: Our system takes user music and notification text as input, pre-processes them to extract information, and uses this information to generate new melodies and melodic notifications that can then be integrated into the current song.

music transformer to account for music and text prosody, and (2) a module that can automatically segment the syllables in spoken text and map each syllable to a melody note. Our main goal is to improve user experience by ensuring that the output voice messages are intelligible and blend well with the current song, minimizing intrusiveness and interruptions to music listening.

3.1 Input and pre-processing

Our method requires the pre-processing of user music and notification text to extract essential information required for melody generation and voice modification.

3.1.1 Symbolic music information. Our method assumes access to a symbolic representation of the listener's music, specifically, the melody notes, chords (harmony), and the click track (beats and tempo). Currently, we retrieve this information for specific pop songs from Hooktheory's TheoryTab database, which contains manually labeled annotations for nearly 50k songs. For songs that are not documented in this database, it is possible to automatically transcribe (predict) this type of symbolic music information from audio [16, 18, 58], albeit with less precision. We focus on the retrieval setting primarily because we aim to create musical notifications, rather than music transcription. Additionally, we envision that in a real-world implementation of our system, artists could be given control over how their music should and shouldn't be modified for musical notifications by providing symbolic information and additional usage metadata, as discussed in Section 7.

3.1.2 Synthesizing the text transcript. A key requirement of our approach is that the musical notifications are intelligible. To achieve this, we modify the audio outputs of text-to-speech (TTS) systems to create a singing voice synthesis system with high intelligibility. From an input text notification to our system (e. g., "remember to take your meds"), we use an off-the-shelf TTS system [20] to synthesize the text as speech audio. Then, we input the speech audio and the text transcript to an off-the-shelf forced alignment system [59] to estimate the onset time of phonemes in audio. Finally, as visualized in Figure 3, we group phonemes into syllables by filtering through vowels, ensuring that only one vowel was present in each cluster of phonemes, yielding an ascending list of syllable timestamps $[t_1, \ldots, t_L]$. Here, L is the number of syllables in the original text transcript, and t_i is the estimated onset time of syllable i. To remove initial silence, we shift all timestamps and the audio by a constant amount of time, such that $t_1 = 0$. This list of syllable timestamps will later be used in both melody generation and voice modification steps of the system.

3.2 Generating new melodies

We contribute a method for generating new melodies that fit with both the detected or retrieved musical context, as well as the natural spoken rhythm of the notification text. The relationship between musical melody and language is a complex one [6, 15], and it may be the case that there is no suitable part of the original melody that fits with the new text. Hence, to produce natural-sounding results, we propose to *generate* melodies with awareness of both the musical context and the notification text.

3.2.1 Background: the Anticipatory Music Transformer. Our proposed method is based on the Anticipatory Music Transformer [49], a large language model (LM) capable of symbolic music generation. Most symbolic music LMs generate notes in a left-to-right fashion,

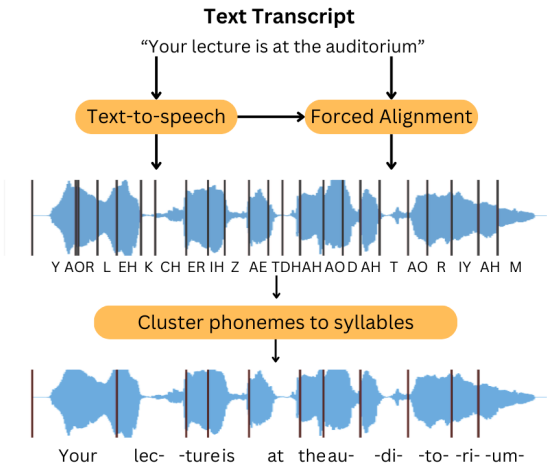

Figure 3: Given a text transcript, our method estimates musically relevant prosody information by first synthesizing that text as audio with TTS and then estimating onset times for each syllable in the audio. To get syllable onset times, we first use an off-the-shelf forced alignment method [59] to estimate onset times for individual phonemes in the TTS output, and then cluster phonemes into syllables.

predicting subsequent notes from past ones. However, in our setting, we hope to insert a generated melody into the middle of an existing one, necessitating awareness of not only past notes but also future ones. The Anticipatory Music Transformer offers this capability of generating in the middle of a musical sequence.

More formally, a note e is a tuple $(e^{\text{time}}, e^{\text{dur.}}, e^{\text{ins.}}, e^{\text{pitch}})$, i.e., its absolute start time, duration, instrument category, and musical pitch. The Anticipatory Music Transformer is a probability distribution $P_\theta(\mathbf{e} \mid \mathbf{c})$ over a sequence of notes (the *events*) $\mathbf{e} = [e_i, \ldots, e_N]$, given a disjoint sequence of notes (the *controls*) $\mathbf{c} = [c_i, \ldots, c_M]$. Both sequences are time-ordered, i.e., $e_i^{\text{time}} \leq e_{i+1}^{\text{time}}$, and $c_i^{\text{time}} \leq c_{i+1}^{\text{time}}$.

This setup is meant to facilitate unusually versatile control for music generation, allowing for generating notes from any other sequence of notes (e.g., generating melody from harmony, or generating the past from the future). Accordingly, the events and control sequences can have arbitrary timings (e.g., they can overlap in time, or the controls can come after the events). To make this tractable to model, the Anticipatory Music Transformer adopts a factorization that allows the model to "anticipate" controls some number of seconds δ into the future:

$$P_\theta(\mathbf{e}|\mathbf{c}) = \prod_{i=1}^{N} P_\theta(e_i|\mathbf{e}_{<i}, \mathbf{c}_{\mathcal{I}_i}), \text{ where}$$

$$\mathcal{I}_i = \{j \mid 1 < j \leq M, c_j^{\text{time}} < e_i^{\text{time}} + \delta\}.$$

In both this work and the original paper, $\delta = 5$ seconds.

To model this distribution, the Anticipatory Music Transformer adopts an autoregressive (left-to-right) "decoder-only" Transformer LM [51], similar to the method proposed by Huang et al. [28]. To incorporate anticipation, the controls are shifted forward by δ seconds and then interleaved with the events in time-order (additional details in the original paper), enabling anticipation in a standard Transformer LM. After interleaving, the four attributes in each note tuple are expanded to three sequence timesteps (instrument and pitch are combined into one timestep) and also modeled autoregressively:

$$P_\theta(e_i|\cdot) = P_\theta(e_i^{\text{time}}|\cdot)P_\theta(e_i^{\text{dur.}}|e_i^{\text{time}}, \cdot)P_\theta(e_i^{\text{ins.}}, e_i^{\text{pitch}}|e_i^{\text{time}}, e_i^{\text{dur.}}, \cdot).$$

In the model's vocabulary, 10000 tokens represent discretized start times (10ms intervals up to 100s max), 1000 tokens represent discretized durations (10ms intervals up to 10s max), and 16512 tokens represent the cross product of instrument categories (129) and musical pitches (128).

3.2.2 Generating new melodies for a text transcript. In the following, we detail how to adapt the Anticipatory Music Transformer to generate new melodies for an input text transcript.

Fine-tuning the model. In our setting, the listener's musical context consists of three sequences of note tuples comprising the melody $\mathcal{M}$, harmony $\mathcal{H}$, and click track $\mathcal{C}$. To adapt the Anticipatory Music Transformer to our particular setting, we fine-tune it on a dataset of melody, harmony, and click tracks derived from the Hooktheory dataset [18]. Specifically, for each song in that dataset, we pick a random span in the middle of the melody starting at time t_s and ending at time t_e, and fine tune the pre-trained checkpoint to model $\mathcal{M}_{\geq t_s, <t_e}$ given inputs $\mathcal{M}_{<t_s} \cup \mathcal{M}_{\geq t_e} \cup \mathcal{H} \cup \mathcal{C}$. In other words, when generating the melody in the selected span, the model will be conditioned on all notes from all instruments in the past, as well as all notes from all instruments up to δ seconds into the future. Accomplishing this with anticipation requires configuring controls $\mathbf{c}' = \mathcal{M}_{\geq t_e} \cup \mathcal{H}_{\geq t_s} \cup \mathcal{C}$, and events $\mathbf{e}' = \mathcal{M}_{<t_e} \cup \mathcal{H}_{<t_s}$.

Choosing when to notify. The resulting model is capable of generating new context-aware melodies at arbitrary locations in time, creating a flexible system that could, in theory, generate as soon as possible for higher-urgency notifications, or wait until a more appropriate musical moment for lower-urgency notifications. In our experiments, we examine the latter setting (which we envision as more appropriate for our system), and adopt a simple policy for choosing a musically-appropriate moment: we start at the downbeat of the third measure, and generate a new melody up to two measures in length. Formally, for a song in common time at a tempo of BPM beats per minute, we select $t_s = 8 \cdot \frac{60}{\text{BPM}}$, and $t_e = 16 \cdot \frac{60}{\text{BPM}}$.

Generating one note per syllable. Given the fine-tuned model P_ϕ and target span t_s through t_e, we can generate a new melody $\hat{\mathcal{M}}_{\geq t_s, <t_e}$ by sampling from $P_\phi(\mathbf{e}'_{\geq t_s, <t_e} \mid \mathbf{e}'_{<t_s}, \mathbf{c}')$, using the inference algorithm from the Anticipatory Music Transformer [49]. Taking inspiration from the findings of Johnson et al. that melismatic singing (i. e., stretching syllables to match a melody) is less intelligible [29], we strictly match one syllable to each note to improve intelligibility. To ensure that a sufficient number of notes are generated to convey the text transcript, we reject any samples where the number of notes is less than the number of syllables.

3.2.3 Generating with awareness of prosody. It is generally unlikely that an arbitrary melody pairs naturally with arbitrary text (see Figure 4), even if the number of notes in the melody is equivalent to the number of syllables in the text. For example, try singing the melody of "Twinkle Twinkle Little Star" with the lyrics of the

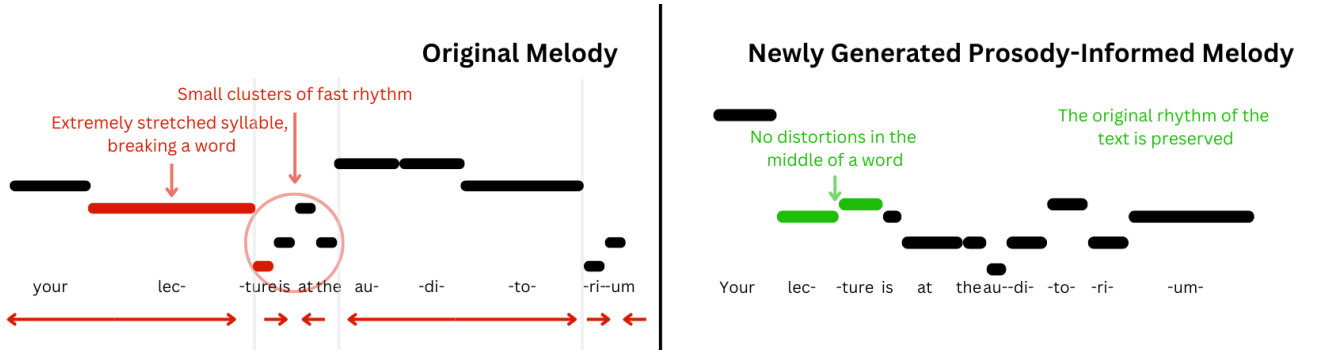

Figure 4: *Left*: When mapping a text transcript to an arbitrary melody, the natural rhythm of the text is broken. Stretching certain syllables for extended durations and compressing some into a short span of time. *Right*: The melody is tailored to the prosody of the text, minimizing any distortions and maintaining the natural flow of speech (examples provided in supplement audio and video).

"Happy Birthday" song—it starts off reasonable but diverges quickly. In preliminary experiments, we found that forcing text to be synthesized to an arbitrary melody (either the original one or a generated one) tended to jeopardize intelligibility.

Here we describe our procedure for generating melodies for text transcripts in a manner that is aware of the natural prosody of the transcript. Given a selected time span t_s and t_e, our goal is to constrain the model to generate a new melody that has one note for each of L syllables in the text notification and is at most $t_e - t_s$ seconds in length. One simple strategy for accomplishing this involves first uniformly stretching the original timings of the synthesized speech. From section 3.1.2, we have the estimated onset timestamps $[t_1 = 0, \ldots, t_L]$ for each syllable in the output of the TTS system. To stretch these to be within the span of t_s and t_e, we define:

$$\hat{t}_i = t_s + t_i \cdot \min\left(D, \frac{D+1}{2}\right), \text{where } D = \frac{t_e - t_s}{t_L}.$$

Intuitively, the first syllable is mapped to the downbeat of the third measure ($\hat{t}_1 = t_s$), and the last syllable will occur no later than the end of the fourth measure ($\hat{t}_L \leq t_e$).

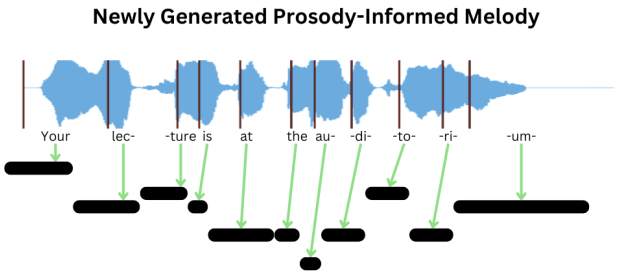

Figure 5: Using the syllable onsets derived from TTS, we generate a new melody that minimizes the distortion of speech rhythm while also considering the new melody's musical fit with user music. We map each syllable to a single musical note.

We leverage the observation that the *prosody* (here, syllable timings, as shown in Figure 5) produced by the TTS system offers us a set of timings under which the text is known to sound natural. Because we're generating new melodies anyway, we can constrain the notes of the generated melody to be within $\pm\alpha$ seconds of the original prosody, i.e., $|e_i^{\text{time}} - \hat{t}_i| < \alpha$. We set $\alpha = \frac{60}{4 \cdot \text{BPM}}$, i.e., the length of one sixteenth note. This tolerance factor ensures that the syllable timings of the generated melody are close to the original ones while giving the model some flexibility to make the timings a bit more musically rhythmic. To accomplish this, we define the **prosody-aware generation** of melodies as sampling from a model with two inference-time constraints that respectively adjust e_i^{time} (the note start) and $e_i^{\text{dur.}}$ (the note duration) of melody note i with respect to syllable onset time $\hat{t}_i$:

$$P'_\phi(e_i^{\text{time}} \mid \cdot) \propto \begin{cases} P_\phi(e_i^{\text{time}} \mid \cdot) & \text{if } |e_i^{\text{time}} - \hat{t}_i| < \alpha \\ 0 & \text{otherwise,} \end{cases}$$

$$P'_\phi(e_i^{\text{dur.}} \mid e_i^{\text{time}}, \cdot) \propto \begin{cases} P_\phi(e_i^{\text{dur.}} \mid e_i^{\text{time}}, \cdot) & \text{if } |e_i^{\text{dur.}} - (\hat{t}_{i+1} - \hat{t}_i)| < \alpha \\ 0 & \text{otherwise.} \end{cases}$$

This setup does ensure that we generate precisely the same number of notes as syllables L, but does not ensure that generated melody notes are non-overlapping. Accordingly, after we've sampled the new melody $\hat{\mathcal{M}}_{\geq t_s, < t_e}$, we postprocess it to set $e_i^{\text{dur.}} = e_{i+1}^{\text{time}} - e_i^{\text{time}}$.

3.3 Musical voice synthesis

To synthesize the final audio, we propose a novel method of modifying TTS outputs to match the pitch and duration of a melody (generated or otherwise) in MIDI format. We take the same TTS output used to extract natural prosody onset timings in Section 3.1.2 and use the syllable onset timings extracted to further modify the speech signal. After obtaining the start times of each syllable in a given speech audio clip, we proceed to remap the pitch and duration of each syllable so that they match the generated melody. We chose this speech-modification based approach over direct singing voice synthesis (SVS) after preliminary experiments with off-the-shelf SVS systems showed significant intelligibility issues.

To achieve this remapping, we use a digital signal processing technique known as Time-Domain Pitch Synchronous Overlap and Add (TD-PSOLA) [37, 38]. This technique operates by taking as input the original audio, a list of onsets present in the original audio, a corresponding list of target onsets intended for time stretching the audio, and a list of fundamental pitches intended for pitch shifting the audio. It then processes this input data to generate a modified version of the audio. In this altered version, both pitch and duration are adjusted to align with the specified input lists.

Due to a hard constraint on the fundamental pitch, the resulting pitch-shifted speech audio may resemble the outputs of commercial vocal pitch-correction software such as Melodyne [24] and Autotune [48]. However, our implementation differs from these tools with the addition of syllable segmentation and automatic mapping of the segmented syllables to MIDI input, which are not available features in existing tools. We see potential applications of this automated modification process to also be useful in the creation of speech-musification content, such as the works of songify the news [44] and MAD/"guichu" artists [14, 57].

3.4 Integration into music

As a final step, we take the musical notification and integrate it into the user's current music at the target location (Section 3.2.2). For target locations *without* vocals, we simply overlay the speech output at the desired temporal location and slightly decrease the volume of the track. Unlike conventional virtual assistants where the music volume is severely reduced, we only slightly attenuate the music volume so that the overall amplitude does not distort or clip when the musical notification is mixed in. For target locations *with* vocal content, we first use Spleeter [26] to separate the audio into vocals and instrumental accompaniment. Then, we completely replace the original vocals in the target location with the musical notification, and additionally slightly attenuate the instrumental accompaniment. To create the final output, we currently mix components together manually in a commercial digital audio workstation, allowing us to refine small alignment issues between the retrieved symbolic music and the original audio. However, this refinement procedure could be automated in the future [18].

4 TECHNICAL EVALUATION

We conducted a technical evaluation to investigate whether our approach generates intelligible voice messages. Specifically, we evaluated each audio clip using an ASR (Automatic Speech Recognition) model, then compared the output transcript with ground truth and calculated the word error rate (WER). We used OpenAI's whisper (medium, English-only model) for speech recognition and JiWER 3.0.3 for WER calculation. We test our method and ablate the individual subcomponents, traditional singing voice synthesis with varying parameters, and conventional (non-melodic) TTS.

4.1 Procedure

We selected a pool of 10 different pop songs excerpts (e. g., *Numb* by Linkin Park; *Take on Me* by A-ha) each mapped to a different input text (e. g., "Project report due at 3pm today"; "New calendar invite"). The full list of songs and texts can be found in Appendix A.1.

We examine intelligibility under three different sources of melody:

- **Original melody.** The original melody and chords of pop songs directly retrieved from the TheoryTab database. Since the number of syllables in the selected measures may not match with input text, we truncate the melody to match the syllable count.
- **Generated.** Melody replaced by the output of the generation procedure described in Section 3.2.2.
- **Generated with prosody awareness.** Melody replaced by the output of the generation procedure described in Section 3.2.3, i.e., where the note timings are additionally constrained by prosody information estimated in Section 3.1.2.

Each melody is input to both commercial state-of-the-art SVS software (ACE Studio [4]), and our speech-to-melody mapping module to synthesize melodic voice from TTS. This resulted in a total of 70 audio clips, i. e., 10 audio clips × 3 generation method (original, generated, generated with prosody) × 2 voice synthesis (SVS, TTS) + 10 baseline audio clips (TTS without melody). For each audio clip, we compute the WER value. We process each transcript to remove punctuation, set every word to lowercase, and convert numbers to text (represent "5" as "five").

4.2 Results

All WER values are detailed in Table 1. Unmodified TTS had a low WER rate at 2%. In general, samples generated using our speech-to-melody method (pitched TTS) exhibited significantly lower WER (mean of 10%) than clips generated using SVS (mean of 33%). This matches the expectations that singing voice synthesis is not optimized for high intelligibility. Using prosody-aware generation with pitched TTS showed the lowest WER from all melodic voice generation methods, highlighting that it can successfully produce intelligible voice messages. Results also indicate no meaningful difference in WER between original and generated melodies for pitched TTS.

Through the results of the technical evaluation, we decided that current commercial SVS software is not sufficient to convey information intelligibly and that directly using the original melody of the song will likely result in major distortions to the original prosody of the speech, impacting intelligibility. For all further evaluations, we use our best-performing method, specifically prosody-constrained melody generation with pitched TTS.

Synthesis	Melody	WER	Music-aware
TTS	N/A	2%	×
Pitched TTS	Original	13%	✓
	Generated	15%	✓
	+Prosody-aware	**2**%	✓
SVS	Original	38%	✓
	Generated	32%	✓
	+Prosody-aware	30%	✓

Table 1: Mean WER for different voice synthesis and melody generation methods.

5 USER STUDY

We conducted a study with end users to gain insights into the usage of our current implementation of musical voice assistants. 12 participants experienced speech messages integrated into popular songs using our system, as well as a baseline of non-musical text-to-speech outputs. We analyze users' preferences via subjective ratings and qualitative comments.

5.1 Study design

Participants were asked to perform everyday work on their own personal laptops while sitting in a typical open space office. At the same time, they listened to eight songs in total, each of which contained one spoken notification created using two separate methods. As a baseline, voice notifications were delivered using Google's text-to-speech system. For the musically modified speed condition, we used a combination of prosody-constrained melody generation, Pitched TTS, and singing voice conversion, as described earlier. Each method was used in four songs.

Audio clip preparation. We used the same 10 songs and text from the ASR evaluation (full list in Table A.1). The songs are selected out of the "top 50 songs" on the theoryTab database [27], and the texts are arbitrary to cover a range of possible speech notifications. While the majority of songs on the top songs list were pop and dance music, we selected wide coverage of genre (e. g., nu-metal, retro chiptune, classical, psychedelic jazz), key (9 represented), year (1680 - 2017), tempo (82-169 BPM, $M = 116.4$, and $SD = 24.16$), and the selected section for integration (5 Verse, 3 Chorus, 2 Pre-chorus). We control the timing of the experiment by trimming songs (gradual fade out) to be around 3 minutes or less in duration. All songs were embedded with exactly one notification, at the 3rd measure of the specified section of integration. Non-modified speech is integrated at the same time as their counterparts but has a randomized offset applied to more accurately represent notifications not entering on downbeats.

For each participant, we randomly selected 8 unique songs out of the pool, presented in a randomized order. Each participant first listened to four songs with notifications from one method, and then four songs with notifications from the other method, counterbalanced by alternating which method was presented first.

SVC to improve naturalness of vocal timbre. After an initial pilot study, a common critique was the robotic and unnatural vocal timbre. To address this issue, we decided to implement singing voice conversion (SVC) techniques to make the vocals sound more human. Specifically, we employed SoftVC VITS 4.0 [47]. Our goal is not to mimic any particular artist, but rather to enhance the naturalness of the final output. However, due to the current limitations of SVC communities and model hosting platforms [22], the available models we found are typically based on existing music artists or celebrities. As a workaround, we selected a voice model trained on the data of a female pop artist whose songs did not overlap with those in our evaluation set. Moving forward, we intend to develop our own SVC model or an intelligible singing voice synthesis (SVS) model using publicly available datasets. The SVC model takes the pitched TTS as input and outputs a voice messages that is similar in melody but aims to be more natural in timbre.

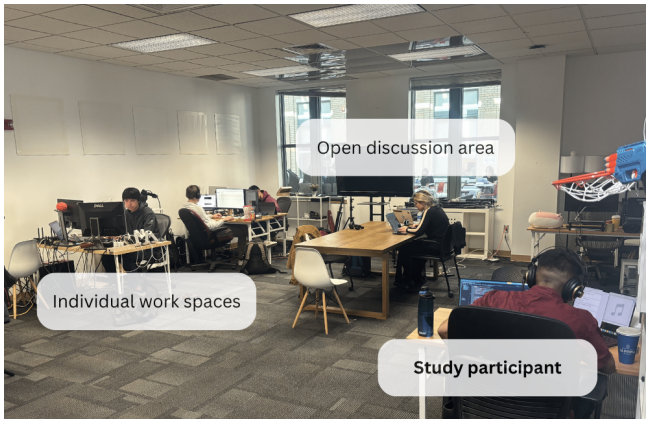

Figure 6: The office space used for user studies. Multiple conversations happen in this space with various volumes.

5.2 Participants and apparatus

We recruited 12 participants (5 female, 7 male, age: $M = 24.33$ years, $SD = 4.75$), all students and staff from a local university. Participants listened to music on a regular basis (weekly listening: 4 × 10+ hours, 6 × 3 - 10 hours, 2 × 1-3 hours) during activities such as commuting (n = 11), studying (n = 10), exercising (n = 10), and doing housework (n = 11). Many participants reported having some experience in music (8 intermediate, 2 novice, 2 professional), and were fluent in English (8 native/bilingual, 3 full professional, 1 professional working).

Participants performed the study on a desk in a busy, but not overwhelming, office environment where multiple groups of workers were chatting at various volumes, shown in Figure 6. We selected this environment over a controlled, quiet space to simulate the amount of noise that users may experience in real-life usage. The audio was presented through a pair of AKG K240 Studio semi-open (does not block outside noise) headphones. Songs and notifications were played back on a separate computer. All participants were compensated with a $15 Amazon gift card.

5.3 Procedure

After a brief introduction, participants gave informed consent and completed a demographic questionnaire. They then calibrated the volume of the headphones to a comfortable level. While performing their personal tasks, participants experience all eight songs with embedded notifications. Whenever they encountered a notification, they were asked to transcribe the message on a separate computer we provided. At the end of four songs for one method, they completed a brief questionnaire. After both sets of songs, we conducted semi-structured interviews, gathering insights into users' preferences, as well as the benefits and challenges of both methods.

5.3.1 Data collection. To best simulate realistic usage, we did not disclose to participants how often the notifications are played. For each notification, participants were asked to record the timestamp and transcribe the message. At the end of each 4 song block, we asked participants to subjectively rate the notifications for noticeability ("I immediately noticed the message"), clarity ("I clearly understood the message"), harmonicity ("The message fits well

with the current music"), intrusiveness ("The message felt intrusive to my music listening experience"), enjoyment ("The message was presented in a delightful way"), and overall user experience ("Overall, my experience as a user was good"); all on a scale from 1 (strongly disagree) to 7 (strongly agree). We analyzed the subjective ratings for statistically significant differences using paired-samples sign test, performed in IBM SPSS Statistics 29. Comments from the semi-structured interviews were grouped and analyzed using open and axial coding methods.

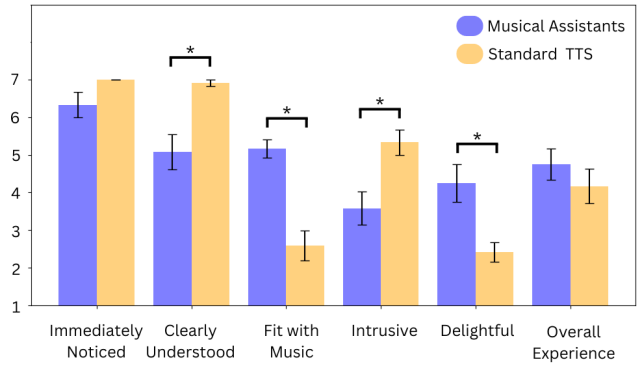

Figure 7: Mean ratings on a scale of 1 (strongly disagree) to 7 (strongly agree). Error bars reflect standard errors, and asterisks (*) imply statistically significant differences.

5.4 Results

5.4.1 Subjective ratings. Analyzing rating data with paired-samples sign tests (Figure 7), we found that while the participant transcriptions were generally correct, our approach is still subjectively perceived to be less clearly understood ($p = 0.004$). However, participants rated our approach to have a better fit with music ($p = 0.006$), less intrusive ($p < 0.001$), and more delightful ($p = 0.021$) compared to the baseline. Similar opinions are expressed in written feedback and during the interview.

5.4.2 Notification intelligibility. Only one participant missed one notification (musical condition) across all 88 notifications (44 per condition across all participants). We manually compared each participant's transcriptions of the message to the ground truth text, finding minor errors in four messages of the musical condition and in two messages in the TTS version. In general, participants understood the notification correctly, even though the exact wording sometimes differed slightly from the ground truth. One particular song saw many errors such as "check report by 5 today" and "Project due by tonight Friday" (correct: "Project report due by 5 today"). Participant P11 specifically commented on this during their interview, stating that this song in particular uses a synthesized voice that merged too well with our synthesized speech. By the time they noticed that this was not part of the song, it was hard to parse out the exact first half of the message. Aside from this rare case of alignment in the usage of synthesized voice, comments and the subjective ratings indicated that intelligibility was high for both methods.

5.4.3 Qualitative feedback. In the following, we summarize the key findings from the semi-structured interview.

Music-aware notifications. All participants could clearly distinguish between conventional speech messages and our modified musical speech messages. Participants described the baseline condition as being similar to regular TTS, or commercial voice assistants like Apple Siri. Participants described our approach as matching the music in terms of rhythm and pitch (other terms used: beat, key, melody, pace, flow, etc), singing over the music, and some specifically mentioned the resemblance with autotune ($n = 3$).

Better blend and less interruption. Most participants found the baseline condition to be disruptive to their music listening experience and that the modified version blends better with music, making it less intrusive ($n = 11$). Many participants found the musical voice to also be distracting as its vocal timbre did not match the style of the music, but still less distracting than cutting out the music ($n = 7$). Better blend, however, may also entail additional mental processing to understand the information. P6 wrote in their open-ended response that "The messages were clear in terms of understanding, but it being so similar to the music required me to process what the message was saying for slightly longer than usual." Both P3 and P11 found that understanding the message was most challenging in the first half. They recommended adding space or tone before the notification to signal the start of the message more clearly.

Open to modifications. Participants were generally familiar with the music they listened to during the study due to the song popularity. Contrary to our initial speculations, participants noted that they do not feel strongly about the modified version having a different melody. They were open to the idea of introducing new musical elements into existing songs, especially if it is for the sake of receiving important notifications ($n = 9$). Both P10 and P11 noted that they did not even realize that the melody was different from the original song since it matched so well with the music.

Participants value intelligibility, continuity, and context-sensitivity. When asked what they value in music voice assistants, participants prioritized clarity ($n = 10$) and continuity ($n = 8$). They want clear and distinct notifications that blend seamlessly with music to minimize distractions. P2 wrote in their response that they "absolutely hate missing notifications, but also don't like my music being interrupted by weird Text-to-Speech voices, which ruin the listening experience." Many also highlighted the importance of timing, that they want the voice messages to be concise, and to take message importance into account ($n = 5$). P1 and P2 expressed concerns about the system's effectiveness when users are in more attention-demanding tasks, indicating a need for granular adaptivity in how the notification is integrated.

Improving vocal timbre and intelligibility. Participants suggest that the notifications should more closely match the genre or the original artist and be overall more natural ($n = 8$). P10 proposed that the classical song presented to them during the study could benefit from an opera singing style, whereas the rock song presented to them could have a rock voice. Some participants expressed difficulties understanding the modified condition ($n = 8$) while the rest thought they were perfectly intelligible.

Figure 8: *Left*: User is engaged in sport activities while receiving a reminder that an assignment is due later today. *Center*: While working on circuitry, the user receives a new meeting invite from his coworker. *Right*: Preparing coffee as part of the user's morning routine, they are reminded that trash pick-up is scheduled today.

General usage. Despite the varying opinions on the effectiveness of the modified notification system, all participants stated that they would prefer to use it over the conventional approach, especially if improvements were made to enhance clarity and singing timbre. Participants generally found the concept of blending notifications with music for a better listening experience promising. Some participants openly expressed their disdain for the interruptions of conventional speech notifications. They either always disable them, use them only when absolutely necessary, or desire to disable them but are unsure how to do so on their own devices ($n = 5$).

6 SCENARIOS OF USAGE

We imagine our implementation of musically embedding speech notifications can improve user experience in a variety of scenarios, such as those depicted in Figure 8. Any situation where the user is listening to music can be an opportunity for musical assistants to intervene.

Exercising. In a dynamic squash practice session, the user is fully immersed in the activity while enjoying his high-energy workout playlist. Because their smartphone is stored outside the court, they becomes oblivious to notifications, leading them to completely forget about the report due later today. Without interrupting the pace of their beats, the musical assistant sings a friendly reminder to remind the user of their academic duties.

Prototyping electronics. To better stay concentrated on their circuitry crafting task, an activity that the user treasures for the serene experience, the user puts on instrumental jazz with a calming flute solo. Without breaking the user's focus, the musical voice assistant notifies them of a new meeting invitation from their coworker. The user stops their current work at a natural stopping point, promptly opens their laptop nearby, and joins the discussion.

Morning coffee. Tired from a party last night, the user starts their morning with a cup of coffee and some catchy pop tunes to help wake them up. Still drowsy, the user completely forgets that trash pickup is scheduled for this morning. With the help of musical assistants, this important information is delivered to the user without any abrupt stops in music to further disorient them. Processing this information at a comfortable and self-determined pace, the user leisurely brings the trash outside and gets ready for another productive day.

Airport. The user reads a book and listens to music while waiting to board a flight. Their noise-canceling headphones ensure a pleasant listening experience even in this crowded and noisy airport lounge. However, noise cancellation also makes them less aware of their surroundings and important announcements, pressuring them to check for boarding activities every few minutes. With the help of musical assistants, the user can remain focused on their book, knowing that any announcements will be gently presented.

7 DISCUSSION

In the following, we discuss the benefits and limitations of our approach, and outline directions for further research.

7.1 Intelligibility and timbre

A key limitation of our current implementation is the intelligibility and unnatural timbre of the synthesized voice, which is largely caused by the hard pitch constraint imposed by the TD-PSOLA method during pitch stretching. We acknowledge that this synthesis method is not ideal and hope to explore alternative approaches in the future. Generating singing voices that are both natural and intelligible is a key challenge in computer music research, and researchers have suggested that training data is a key bottleneck [13]. Ideally, with the creation of high-quality singing voice datasets optimized for intelligibility, we can train SVS models that are highly intelligible and further improve user experience. Given improved

SVS models, replacing our current voice synthesis method is simple, as it is fully decoupled from the melody generation component in our implementation. We also hope to explore the synthesis of artist-specific voices in the future, to better blend the notification as seamlessly as possible.

7.1.1 Notification saliency. Challenges with intelligibility can arise if notifications are presented too subtly. For such notifications, participants reported that trying to understand the message required extra cognitive effort. One participant noted that they did not realize that the spoken notification was not part of the song until halfway through the notification, making them miss the first part of the message. This finding falls in line with our work on musically-integrated ringtones [52], where seamlessly integrated notifications were not necessarily the most preferred, since catching them required extra cognitive effort and could cause false positives in notification detection. In that study, participants preferred a medium-urgency parameter setting that is musically integrated but not overly subtle. We believe that the same can be applied to our approach for sung notifications. One suggestion from participants was to add an auditory cue preceding the message, a design already deployed by existing virtual assistants such as Apple Siri. This approach could improve the noticeability of seamlessly integrated notifications. In the future, we plan to explore these additional design possibilities, quantify the cognitive effort associated with them, and compare our approach to those crafted by human musicians.

7.2 Granularity and context-sensitivity

One recurring comment from participants was the importance of keeping messages brief and appropriate for context. For example, notifications should not read out entire promotional emails. We believe that these nuanced understandings of how and when notifications are received highlight the complementary nature of our work. Our approach of music integration may serve as an extra layer of granularity in the presentation of such messages, as opposed to replacing conventional notification delivery completely. Low-priority notifications, which users wouldn't mind missing, could seamlessly blend into the music. Medium-priority notifications might employ a slightly altered melody or vocal timbre to distinguish them. High-priority notifications, on the other hand, could use conventional text-to-speech (TTS) methods with volume ducking for prominence. Additionally, an interesting approach to ensure message brevity is to leverage LLMs to paraphrase long messages into syllable counts or prosody rhythms that best matches the current song, while balancing for the level of detail best suited for different types of messages.

7.3 Alternative integration methods

Our current integration approach involves synthesizing a melodic voice to replace the lyrics of the original song, which may be most appropriate for integration with Western pop music. However, our approach is less relevant to broader musical traditions that may use entirely different musical tuning systems or convey language through different vocal styles and techniques. For instance, hip-hop utilizes rapping, gospel relies on backing harmony vocals, and dance music incorporates pre-drop chants, among others. Similarly, while our current approach can be applied to any part of a song, we recognize the importance of selecting optimal moments for notification delivery. For example, rather than interrupting a high-energy chorus, we aim to identify sections with less action and more space, potentially as a function of notification urgency. Exploring suitable modes and opportune moments for integration on a genre-by-genre or song-by-song basis could further enhance the seamless integration of speech notifications.

7.4 Artist-in-the-loop: Authoring adaptations as creative process

We believe that artists should have agency over how their work is presented and that more discussion about the ethical implications of generative audio models is needed, as highlighted by prior work [3]. Any commercially available music is already subject to playback and modification in contexts that may deviate from the artist's vision or intent, for example through being interrupted by spoken notifications, or through third-party remixing. We imagine that artists prefer more control over how their work is modified. Our work offers an additional dimension for this exploration.

We believe that artists should have full control over whether and which modifications are allowed. We envision an opt-in ecosystem of music-aware assistants where artists can elect to actively participate in shaping how their music is altered. This involvement could extend to both allowing alterations to their music, and contributing to the sound design of these adaptations. For instance, artists could annotate sections of their songs that are best suited for various types of alterations, and even compose new and modified sections specifically designed to integrate seamlessly with external elements such as speech notifications. Artists may also elect to have information presented using models trained on their own voice, as opposed to a generic voice which we explore here. We imagine virtual assistant providers could offer tools to artists that would allow them to interactively simulate the outcomes of their annotations with random notification text. By engaging artists in this process in future work, we aim to create a collaborative environment where music can dynamically adapt to diverse contexts without compromising the artistic integrity and vision of the creators.

Similarly, we expect the same level of agency to be desirable for end users. They should be able to opt in/out of any kind of notification delivery, be it integrated or not, including for individual songs that they may have a special attachment to. Future work should explore ways to enable this control, as well as ways to implicitly infer when a notification is appropriate to reduce users' burden.

8 CONCLUSION

We present the concept of and method for musical assistants, along with a novel melody generation technique that optimizes voice messages for both intelligibility and musicality. We believe that integrating voice messages into users' music is an interesting direction for personalized virtual assistants that aim to be well integrated in users' tasks, and not distract them or draw attention away from their main tasks. Through a user study, we found that users are receptive to the idea of integrating speech notifications into their music and values the continuity and unobtrusiveness brought by this technology. While our current melody generation component

is successful in supporting musical voice assistant interactions, the voice synthesis component is still lacking in terms of intelligibility and genre-appropriate vocal timbre, which impacted the way it was perceived by users. In the future, we wish to further explore the possibilities of improving intelligibility in SVS models and whether this will strongly boost user experience. As with visual notifications, we believe that the design of auditory interactions should be context-sensitive, personalized, and should aim at being well integrated and beneficial for users. We see our work as a step towards expanding this design space.

REFERENCES

[1] Tawfiq Ammari, Jofish Kaye, Janice Y. Tsai, and Frank Bentley. 2019. Music, Search, and IoT: How People (Really) Use Voice Assistants. *ACM Trans. Comput.-Hum. Interact.* 26, 3, Article 17 (apr 2019), 28 pages. https://doi.org/10.1145/3311956

[2] Ishwarya Ananthabhotla and Joseph A Paradiso. 2018. SoundSignaling: Realtime, stylistic modification of a personal music corpus for information delivery. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 4 (2018), 1–23.

[3] Julia Barnett. 2023. The ethical implications of generative audio models: A systematic literature review. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*. 146–161.

[4] Ltd Beijing Timedomain Technology Co. 2024. ACE Studio | Create Limitless AI Vocals. https://acestudio.ai/.

[5] 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.

[6] Richard Burleson. 1992. Functional Relationships of Language and Music: The Two-Profile View of Text Disposition. *La Linguistique* 28, 2 (1992), 49–63. http://www.jstor.org/stable/30248666

[7] 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

[8] Kyoyun Choi, Jonggwon Park, Wan Heo, Sungwook Jeon, and Jonghun Park. 2021. Chord conditioned melody generation with transformer based decoders. *IEEE Access* 9 (2021), 42071–42080.

[9] Lauren B Collister and David Huron. 2008. Comparison of word intelligibility in spoken and sung phrases. (2008).

[10] Nathaniel Condit-Schultz and David Huron. 2015. Catching the lyrics: Intelligibility in twelve song genres. *Music Perception: An Interdisciplinary Journal* 32, 5 (2015), 470–483.

[11] Benjamin R Cowan, Nadia Pantidi, David Coyle, Kellie Morrissey, Peter Clarke, Sara Al-Shehri, David Earley, and Natasha Bandeira. 2017. " What can i help you with?" infrequent users' experiences of intelligent personal assistants. In *Proceedings of the 19th international conference on human-computer interaction with mobile devices and services*. 1–12.

[12] Anne Cutler and D Robert Ladd. 2013. *Prosody: Models and measurements*. Vol. 14. Springer Science & Business Media.

[13] Shuqi Dai, Siqi Chen, Yuxuan Wu, Ruxin Diao, Roy Huang, and Roger B Dannenberg. 2023. Singstyle111: A multilingual singing dataset with style transfer. In *in Proc. of the 24th Int. Society for Music Information Retrieval Conf*, Vol. 1. 4–2.

[14] Kendrick Davis. 2018. *The Manic Mashups Charming China's Internet.* https://www.sixthtone.com/news/1003118 Accessed: April 1, 2024.

[15] Diana Deutsch, Rachael Lapidis, and Trevor Henthorn. 2008. The speech-to-song illusion. *J. Acoust. Soc. Am* 124, 2471 (2008), 10–1121.

[16] Chris Donahue and Percy Liang. 2021. Sheet sage: Lead sheets from music audio. *Proc. ISMIR Late-Breaking and Demo* (2021).

[17] Chris Donahue, Huanru Henry Mao, Yiting Ethan Li, Garrison W Cottrell, and Julian McAuley. 2019. LakhNES: Improving multi-instrumental music generation with cross-domain pre-training. *arXiv preprint arXiv:1907.04868* (2019).

[18] Chris Donahue, John Thickstun, and Percy Liang. 2022. Melody transcription via generative pre-training. *arXiv preprint arXiv:2212.01884* (2022).

[19] Jeff Donahue, Sander Dieleman, Mikołaj Bińkowski, Erich Elsen, and Karen Simonyan. 2020. End-to-end adversarial text-to-speech. *arXiv preprint arXiv:2006.03575* (2020).

[20] Pierre Nicolas Durette. 2024. gTTS. https://pypi.org/project/gTTS/. Version 2.5.1.

[21] Gavin Edwards. 1995. *Scuse Me While I Kiss This Guy.* Simon and Schuster.

[22] Michael Feffer, Zachary C. Lipton, and Chris Donahue. 2023. DeepDrake ft. BTS-GAN and TayloRVC: An Exploratory Analysis of Musical Deepfakes and Hosting Platforms. In *HCMIR*.

[23] William W Gaver. 1989. The SonicFinder: An interface that uses auditory icons. *Human–Computer Interaction* 4, 1 (1989), 67–94.

[24] Celemony Software GmbH. 2024. Celemony - What is Melodyne? https://www.celemony.com/en/melodyne/what-is-melodyne.

[25] Florian Heller and Johannes Schöning. 2018. Navigatone: Seamlessly embedding navigation cues in mobile music listening. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–7.

[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] Hooktheory. [n. d.]. Top 50 Songs - Hooktheory. https://www.hooktheory.com/theorytab/charts/chart/top. Accessed March 29 2024.

[28] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. 2018. Music transformer. *arXiv preprint arXiv:1809.04281* (2018).

[29] Randolph B Johnson, David Huron, and Lauren Collister. 2014. Music and lyrics interactions and their influence on recognition of sung words: An investigation of word frequency, rhyme, metric stress, vocal timbre, melisma, and repetition priming. *Empirical Musicology Review* 9, 1 (2014), 2–20.

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

[31] 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*. 118–133.

[32] Eugene Kharitonov, Damien Vincent, Zalán Borsos, Raphaël Marinier, Sertan Girgin, Olivier Pietquin, Matt Sharifi, Marco Tagliasacchi, and Neil Zeghidour. 2023. Speak, read and prompt: High-fidelity text-to-speech with minimal supervision. *Transactions of the Association for Computational Linguistics* 11 (2023), 1703–1718.

[33] Jaehyeon Kim, Jungil Kong, and Juhee Son. 2021. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *International Conference on Machine Learning*. PMLR, 5530–5540.

[34] Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. 2024. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in neural information processing systems* 36 (2024).

[35] Chien-Feng Liao, Jen-Yu Liu, and Yi-Hsuan Yang. 2022. Karasinger: Score-free singing voice synthesis with vq-vae using mel-spectrograms. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 956–960.

[36] Jinglin Liu, Chengxi Li, Yi Ren, Feiyang Chen, and Zhou Zhao. 2022. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 36. 11020–11028.

[37] Max Morrison. 2021. PSOLA. https://pypi.org/project/psola/. Version 0.0.1.

[38] Eric Moulines and Francis Charpentier. 1990. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech communication* (1990).

[39] Elizabeth D. Mynatt and W. Keith Edwards. 1992. Mapping GUIs to auditory interfaces. In *Proceedings of the 5th Annual ACM Symposium on User Interface Software and Technology* (Monteray, California, USA) *(UIST '92)*. Association for Computing Machinery, New York, NY, USA, 61–70. https://doi.org/10.1145/142621.142629

[40] Adrian C North, David J Hargreaves, and Jon J Hargreaves. 2004. Uses of music in everyday life. *Music perception* 22, 1 (2004), 41–77.

[41] Aaron Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George Driessche, Edward Lockhart, Luis Cobo, Florian Stimberg, et al. 2018. Parallel wavenet: Fast high-fidelity speech synthesis. In *International conference on machine learning*. PMLR, 3918–3926.

[42] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016).

[43] Christine Payne. 2019. MuseNet. openai.com/blog/musenet

[44] schmoyoho. 2019. Very Thin Ice: 10 years of Auto-Tune the News and Songify This. YouTube video. https://www.youtube.com/watch?v=TDf-oYz6hLQ Accessed: April 1, 2024.

[45] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 4779–4783.

[46] Ian Simon and Sageev Oore. 2017. Performance RNN: Generating Music with Expressive Timing and Dynamics. https://magenta.tensorflow.org/performance-rnn.

[47] svc-develop-team. 2023. so-vits-svc. GitHub repository. https://github.com/svc-develop-team/so-vits-svc

[48] Antares Audio Technologies. 1997. Autotune. https://www.antarestech.com/.
[49] John Thickstun, David Hall, Chris Donahue, and Percy Liang. 2023. Anticipatory Music Transformer. *arXiv preprint arXiv:2306.08620* (2023).
[50] Amrita S Tulshan and Sudhir Namdeorao Dhage. 2019. Survey on virtual assistant: Google assistant, siri, cortana, alexa. In *Advances in Signal Processing and Intelligent Recognition Systems: 4th International Symposium SIRS 2018, Bangalore, India, September 19–22, 2018, Revised Selected Papers 4*. Springer, 190–201.
[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
[52] Alexander Wang, Yi Fei Cheng, and David Lindlbauer. 2024. MARingBA: Music-Adaptive Ringtones for Blended Audio Notification Delivery *(CHI '24)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3544548.3581027
[53] Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. 2023. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111* (2023).
[54] 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)*. IEEE, 3735–3739.
[55] Jing Yang and Andreas Roth. [n. d.]. Musical Features Modification for Less Intrusive Delivery of Popular Notification Sounds. In *Proceedings of the 26th International Conference on Auditory Display*.
[56] Yi Yu, Abhishek Srivastava, and Simon Canales. 2021. Conditional lstm-gan for melody generation from lyrics. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 17, 1 (2021), 1–20.
[57] Zach. 2021. *MAD - Know your meme.* https://knowyourmeme.com/memes/mad Accessed: April 1, 2024.
[58] Xinquan Zhou and Alexander Lerch. 2015. Chord detection using deep learning. In *Proceedings of the 16th ISMIR Conference*, Vol. 53. 152.
[59] Jian Zhu, Cong Zhang, and David Jurgens. 2022. Phone-to-audio alignment without text: A Semi-supervised Approach. arXiv:2110.03876 [cs.CL]

A APPENDIX

A.1 List of songs and notifications

List of notifications used in both studies:

(1) Project report due by 5 today
(2) Heavy traffic reported ahead
(3) Don't forget, meeting at 10 PM
(4) Missed call from Mom, call her back
(5) Remember to take your meds
(6) Buy groceries on your way back
(7) Your flight to New York is boarding now
(8) Turn left in 500 feet
(9) New calendar invite
(10) You have 4 unread Slack messages

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

(1) Sad machine, Porter Robinson, Verse
(2) Numb, Linkin Park, Verse
(3) Don't you worry child, Swedish House Mafia, Chorus
(4) Call me maybe, Carly Rae Jepsen, Chorus
(5) Take on me, A-ha, Verse
(6) The legend of Zelda main theme, Koji Kondo, Verse
(7) Get lucky, Daft Punk, Pre-chorus
(8) Wake me up, Avicii, Chorus
(9) Canon in D, Johann Pachelbel, Pre-chorus
(10) Them changes, Thundercat, Verse