Shifting Time Scales: Supporting Live Gesture-Controlled Interactive Generative Music with Speculative Execution

Jason Brent Smith
jason.smith1@northwestern.edu
Northwestern University
Evanston, Illinois, USA

Bryan Pardo
pardo@northwestern.edu
Northwestern University
Evanston, Illinois, USA

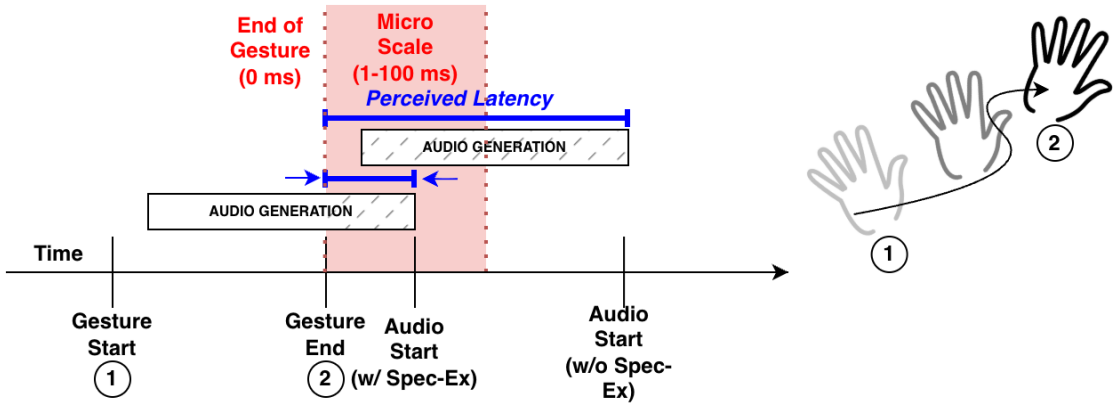

Figure 1: An example of the "shift" in temporal relation between input gesture and output audio that speculative execution (Spec-Ex) can provide. Predicting a completed gesture can reduce perceived latency (blue) in gesture-based systems, which typically begin audio generation after a gesture is completed. Spec-Ex enables the model's output to be used at the *micro* scale (< 100 ms) by generating audio before the gesture completes.

Abstract

Generative AI has enabled the creation of new interfaces for musical expression (NIMEs) that dynamically generate sounds in response to user input. These systems have focused on coarse, text-based instructions delivered at time scales that are not suitable for fine-grained control of sound enabled by conducting-style gestures during a live performance. Additionally, audio generation introduces latency that impedes gesture-based control, limiting the ability of AI-based NIMEs to synchronize musical output with input gestures in real time. This paper presents *Gesture Vocabulary*, an interactive generative music system prototype that uses user-defined hand gestures and motions, such as conducting patterns, as input. This system employs *speculative execution*, generating sound based on predicted future gestures to mitigate latency and produce audio in response to gestures at a time scale suitable for real-time performance within a constrained gesture space. By examining the role of AI in music through the time scales of its interactions with users, we aim to support more expressive performance practices through the embodied control of generative music systems.

Keywords

Generative Music, Artificial Intelligence, Interactive Music, Human-Computer Interaction

NIME '26, June 23–26, 2026, London, UK

1 Introduction

Generative models are one of the most active areas of research in music technology. They have been used to create AI-based interfaces for musical expression that quickly (e.g., several seconds) generate new sounds based on sonic [10, 39], gestural [28], or textual [2] input, but not quickly enough for real-time performance. In this paper, we focus on using gestures to control a generative model in a system suitable for live performance.

Composer and programmer Curtis Roads defines interactions in music-making as a series of **time scales,** ranging from compositional ideas (*supra*), the full length of a piece of music (*macro*), or momentary actions in response to motion and sound (*micro*) [40, 41]. We seek to support artists by broadening the expressive range available to them by introducing a form of control over generative models analogous to that of a conductor over dynamics and tempo. Using gestures, we aim to leverage the physical affordances of human movement to control real-time music generation at the smallest possible time scales. This paper focuses on the *micro* scale (< 100 ms) when discussing how a system responds to motion: the time scale for which a motion is meant to convey a momentary musical change.

In this paper, we **apply the time-scales framework to evaluate the ability of AI-based music performance systems to synchronize with user input**, integrating motion, gesture, and musical decision-making as understood by both human performers and AI agents. We present *Gesture Vocabulary*, an exploratory prototype gesture-controlled generative music system designed to demonstrate how speculative execution can mitigate the inherent latency of diffusion-based audio generation. We evaluate speculative execution in live, interactive generative music performance with respect to perceived latency and prediction accuracy using a simplified, low-dimensional gesture representation.

Time Scale	Musical Decision	
	Traditional Instrument	**AI Generation**
Supra (beyond individual song)	Choice of Instrument	Choice of Model
Macro (length of song)	Tuning, etc.	Delayed LLM Prompts [2]
Meso (1-2 sec)	Adjustments	Instantaneous LLM Prompts [7, 37]
Micro (1-100 ms)	Performance Gesture/Input	Latent control/audio input (RAVE) [6, 10]

Table 1: Roads' time scales and examples of how each is reflected by inputs to traditional and AI-based music performance systems. Arrows indicate potential "shifts" in perspective as a result of applying speculative execution to the user's input.

2 Shifting Perceptions of Latency in Music

2.1 Latency and Synchronization

Low latency is a goal for real-time musical applications [33], and delays can significantly affect performers' effectiveness in collaboration [12, 13]. Traditional instruments (e.g., keyboards) have strict (0-20ms) latency requirements between physical input and sonic output, and changes in latency can alter the "feel of the instrument" [31]. In this paper, we discuss ***perceived latency,*** the difference between the completion of a gesture and the start of the sound generated by a system (see Figure 1).

Delay is a major component of performance analysis for networked [30], distributed immersive [14, 15], and telematic music [4, 17, 38] and can be mitigated in systems designed to accommodate communication times [5, 32]. Low latency enables synchronization of gestures and other inputs with a combined musical output, supporting collaboration [12] and creating a sense of shared movement or synchrony [46]. Synchrony can foster positive attitudes, such as trust and altruism, among human collaborators and between humans and artificial agents [27, 36, 42].

Some generative models have been optimized for real-time performance and operate below Roads' *macro* (1-2 s) scale. The RAVE family of variational autoencoders (VAEs) [6, 10, 11] operates at the "edge of perceivable latency" [10]. However, VAEs use the latent-dimension representation of sound as input and cannot take multimodal inputs that are connected to a shared understanding of the input (e.g., the name of an instrument or the type of rhythm), as diffusion and language models can. As such, there has been a push to speed up these diffusion and language-model approaches. Masked-token modeling [3, 25] has accelerated language-model-style generation, but requires complete audio prompts. Diffusion models like Stable Audio Open Small[1] [37] and MagentaRT[2] are steps towards true interactive music-making based on the semantic meaning of input, but still do not achieve the latency required to *synchronize* with a user's input. These models present a real-time factor (RTF) below 1, generating one second of audio in less than one second. However, to play audio alongside the input with zero perceived latency, the audio must be generated *before* the input ends.

2.2 Gesture and Speculative Execution

Our aim is to control a generative model with gestures as input. The description of gestures plays an important role in linking signal-based descriptions of physical motion with their *meaning* [8, 9]. The use of physical gestures to control musical output has a storied history in the production and use of NIMEs [45]: Michel Waisvisz's The Hands [43] maps sensor data from gloves to sound synthesis parameters. Sidney Fels' Glove-Talk systems [19, 20] use neural networks to map hardware-captured motion gestures to speech-formant sound synthesis, leading to the creation of ForTouch, a NIME that uses these networks alongside a musician's voice for musical performance [21]. Rebecca Fiebrink's Wekinator [22, 23] and collaboration with instrument creator Laetitia Sonami [24] has led to the synergistic development of gestural control systems and a community of users.

Large Language Models (LLMs) have been used to interpret gestures into semantically meaningful text labels. However, these architectures have been shown to be computationally expensive: a prototype zero-shot (operating on gestures not seen in training data) hand-gesture recognition model requires 227 seconds per gesture [47], well outside the *macro* time scale. Pre-trained, vision-based gesture recognition models can operate in real time but are limited to a few gestures, such as sign language numbers [1].

The time to react to a control gesture equals the time required to complete the gesture, recognize it, and generate a response (i.e., the system's actual latency). To reduce the perceived latency of the system (the delay between the end of a gesture and the start of an audio output) without speeding up the model's generation time, we use **Speculative execution (Spec-Ex)**, the process of executing tasks in advance of an expected input [48]. It has been applied within LLMs to rapidly generate text [16, 48] and speech [34]. In our work, we explore the use of Spec-Ex in a live, interactive generative music system. Unlike generative models that use Spec-Ex to speed up the generation *process* [48], we apply it to the recognition of gesture *input* to the generative model, allowing the generation to start before the gesture is complete. As such, our method can be combined with any method that speeds the internal workings of the generative model.

Table 1 links Roads' time scales to decisions made during traditional instrumental performances and in generative AI contexts for live performance. VAE models and exploration of latent spaces allow for *micro*-level sonic adjustments in real time, but require latent spaces to be fully generated before a performance on the *supra* level. Alternatively, LLMs can make decisions in response to real-time input, but the time required to prompt the model limits their use to the *macro* or *meso* levels. We posit that Spec-Ex can take models that are currently useful at Roads' *macro* and *meso* levels and extend their usefulness to the *micro* level by quickly mapping gestures to the control signals (aka conditioning) needed to start generative modeling prior to the completion of the conditioning gesture.Our goal is to use Spec-Ex to make LLM prompting "feel" more like the VAE-based systems' gesture-based control by reducing the perceived latency between input and output. Figure 1 illustrates the effect of gesture prediction on the system's workflow relative to Road's *micro* time scale.

[1] https://huggingface.co/stabilityai/stable-audio-open-small
[2] https://github.com/magenta/magenta-realtime

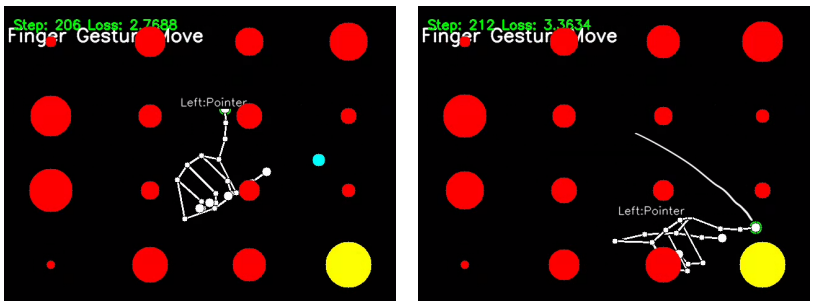

Figure 2: The *Gesture Vocabulary* prototype. Captured hands are represented as wireframes. When a user begins a gesture (left), the system outputs a probability distribution of the gesture's possible ending locations. Higher probabilities are represented by larger circles, with the highest indicated in yellow. It generates audio corresponding with all predictions, in order of likelihood, before the gesture ends (right).

3 System Design

We present *Gesture Vocabulary*, a prototype interactive generative music system that uses speculative execution to predict future gesture inputs and generate prospective musical examples that the user hears as output, alongside the user's completed gesture.

Gesture Vocabulary is a Python application, with a workflow depicted in Figure 3. A user performs hand gestures in front of a camera (Figure 2), which is continuously processed with OpenCV and Google MediaPipe [49] to detect the human body in the input video. Using a keyboard, the user defines and records gesture instances to retrain the gesture classification model. While a user performs motions, their gestures are classified in real time using neural networks to identify the hand sign (the shape of the user's hand) and motion (movement across multiple frames). The resulting classifications are then used to prompt an audio generation model. The system uses separate threads for gesture capture and classification, model training, and audio generation to avoid interrupting the audio stream.

Gesture Vocabulary, like the MediaPipe-controlled interactive music system *GestAlt* [42], uses two existing neural networks for gesture recognition: a linear network with two hidden layers for hand sign classification and an LSTM-based network for motion pattern classification[3]. It adds two additional neural networks to its workflow. First, the gesture classification output is used to prompt a generative model. Second, to enable a generative music model to output audio in coordination with a gesture, and to counter the inherent delay of the generative model, the system performs *speculative execution*.

[3]https://github.com/kinivi/hand-gesture-recognition-mediapipe

3.1 Integration to a Gesture-to-Music Model

We generate audio with a transformer-based diffusion (DiT) model, Stable Audio Open (SAO) Small [37]. SAO Small is controlled via the textual *meaning* of gestures. The user defines the musical meaning of a gesture using the keyboard. For example, the user could label a gesture as "circular motion" alongside the musical definition of "rapid notes." The user can record examples (typically for 10 seconds), after which the model can be retrained manually over roughly 30 seconds. As with *GestAlt* [42], retraining the model occurs in a separate thread from *Gesture Vocabulary's* camera, audio generation, and speculative execution-based processes; as a result, the system's frame rate and audio output are relatively unaffected while the classifier model retrains. When the system detects that defined gesture, it uses the associated text description to prompt the SAO model. This has inherent latency due to the time required to classify the completed gesture and then generate the audio. In the next section, we discuss how to mitigate this latency by predicting gestures, enabling audio generation to start before gesture completion.

3.2 Speculative Execution

Gesture Vocabulary predicts the completion of a gesture so it can generate audio in time to synchronize with movement. This is easier when fewer data points are available to predict, so the live camera input is decomposed into a lower-dimensional representation of the gesture via dimensionality reduction [44]. Hand sign and motion classification reduce a gesture to four dimensions: a single value representing the hand sign classification, a value for the motion classification, and x-y coordinates.

"Rollback netcode" is used for peer-to-peer synchronization in video games [18, 35] by storing an internal state in a compact data format and simulating future game states to "roll back"

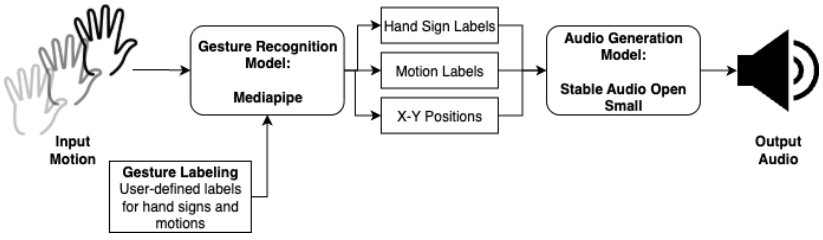

Figure 3: A use-time workflow of *Gesture Vocabulary*. The user performs gestures and labels them to train a Gesture Recognition model; the model's output is used to prompt an Audio Generation model for live playback.

changes caused by untransmitted control data from one player to another. The system renders only the display for the most up-to-date game state. We adapt a similar approach for *Gesture Vocabulary* by predicting multiple gestural patterns and associated musical outputs: The system generates outputs based on each prediction for a gesture, plays the one related to its top choice, and "rolls back" by switching outputs when the predicted and actual gestural inputs mismatch (see Figure 4).

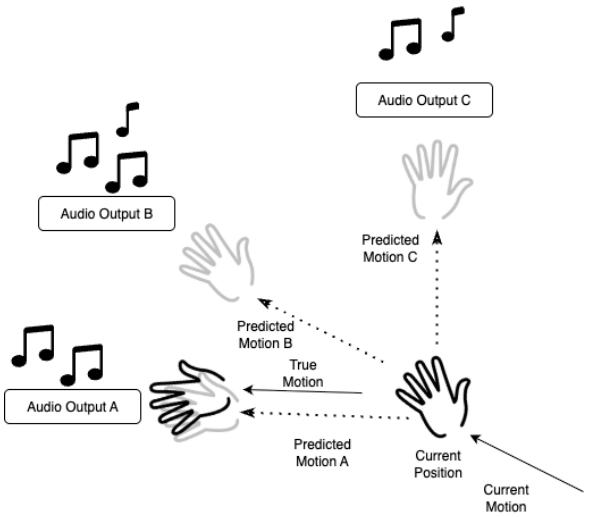

Figure 4: Depiction of *Gesture Vocabulary's* speculative execution process. Audio is generated based on multiple predicted completions of the current gesture. The final musical output is formed by outputting the audio associated with the nearest predicted hand motions.

A typical control gesture that we work with takes roughly 2 to 3 seconds to perform. If the gesture can be recognized within the first 200-300 milliseconds, generation could begin nearly 2 seconds earlier, shifting the perceived control timing from the *macro* to the *micro* level (see Figure 1). *Gesture Vocabulary* uses Spec-Ex to predict a user's *future* gesture inputs. We use a long short-term memory (LSTM) Recurrent Neural Network (RNN) written in PyTorch[4]. This model is **untrained:** it gradually adapts to user input via online machine learning [26], allowing the user to add new gestures during a performance. We also perform a linear spline, using interpolation functions from the scipy library[5] to predict gestures based on a current movement trajectory.

3.3 Constrained Generation

To demonstrate the generation of the full space of possible options for the experiment described in Section 4, the prototype presented in this paper uses a limited set of audio-generation parameters. We constrain the LSTM output to a 4-by-4 grid representing the user's hand location at the end of the gesture. The Stable Audio Open Small model is text-prompted with an instrument name determined by the predicted end location of the hand gesture (see Table 2). On a hardware-limited system, such as a laptop, generating sound files for each possible combination of gesture and motion classification, as well as screen location, simultaneously is not feasible; we constrain the number of generated audio outputs to a feasible set of sixteen options.

[4]https://docs.pytorch.org/docs/stable/generated/torch.nn.LSTM.html
[5]https://scipy.org/

		Column		
Row	trumpet	piano	bass	guitar
	strings	saxophone	harp	trombone
	clarinet	xylophone	violin	tuba
	flute	percussion	oboe	electronic

Table 2: Limited audio generation options of the *Gesture Vocabulary* prototype, with instrument selections determined by hand placement as captured by MediaPipe.

4 Experimental Design

To evaluate the effectiveness of *Gesture Vocabulary's* speculative execution, we analyze the reduction in perceived latency provided by the addition of gesture prediction, as well as the error rate of each prediction made by the system.

Our baseline for comparison is the minimum latency necessary for audio generation. The actual latency of a gesture-to-music model could be calculated as follows:

$$Latency = Duration_{gesture} + Duration_{gesture\_recognition} + Duration_{audio\_generation}.$$

However, we are evaluating perceived latency from the user's perspective. Without knowledge of the model's generative process, the only factor that matters to the user is the difference between their motion and the system's audio output. To measure perceived latency, we use the following equation:

$$Latency_{perceived} = Time_{audio\_start} - Time_{gesture\_end}$$

First, we consider the inference times of the existing models used by the system: MediaPipe [49] and Stable Audio Open Small [37]. On a 2022 Apple M2 laptop, the inference time for hand recognition using MediaPipe and a hand sign classification model averages 25.4 ms, which nearly exceeds a latency threshold that affects a musician's quality assessment of a digital musical instrument [31]. Combined with Stable Audio Open Small's reported latency of 187 ms on a consumer GPU [37], we reach a baseline latency of 212.4 ms. This is also equal to the expected perceived latency, since the system starts playing the audio as soon as it is generated. This exceeds the threshold for Roads' *micro* time scale. As a result, we evaluate *Gesture Vocabulary's* speculative execution in its ability to reduce the 213 ms lag between gesture completion and audio playback towards a zero-second delay.

When *Gesture Vocabulary's* LSTM predicts a gesture, it produces a probability distribution over possible regions of the screen so that it can speculatively generate audio corresponding to any possible gestures (see Table 2). We compare the accuracy of each "Top-N" prediction: Top-1, the model's prediction for the most likely endpoint of the gesture; Top-2, the first- and second-most likely; and so on to Top-16, which includes all possible outcomes. The LSTM model predicts the location of the end of a gesture 3.6 seconds after it begins[6]. We also compare the system against "naive" predictions: a long-term and a short-term linear spline, both based on the assumption that the hand will continue moving linearly for 3.4 s and 220 ms, respectively. These two splines were selected to compare an LSTM's early and final predictions with respect to the time-scale change they represent. The long

[6]70 frames after it begins—given the application's average 20 frames per second, this is roughly 3.6 seconds in the future

spline is intended to generate audio well before gesture completion, modeling the LSTM's initial predictions. The short spline is intended to model predictions that allow the system to generate audio slightly before the gesture ends. We compare two variables:

(1) **Perceived Latency.** The average difference between input gesture completion and audio playback beginning.
(2) **Error Rate.** The percentage of incorrect gestural predictions, measured as whether or not the pre-generated audio output for the actual gestural input also corresponds to the predicted end-of-gesture location.

The system continuously records motion and updates the LSTM with the new data, treating each 30-frame segment as the start of a new gesture. The LSTM predicts the gesture's ending location, which is used to prompt the audio generation model. Each time a gesture is completed, an audio output is played. We compare the predicted gesture data with the actual gesture to determine accuracy, recording the time difference between gesture completion and audio playback as the system's perceived latency. Unlike actual latency, perceived latency can be negative if the audio generation is completed before the gesture. Additionally, the predicted and actual gesture data are used to train the LSTM online. We measure the system's performance for 25 minutes.

5 Results

	LSTM			Spline		No Spec-Ex
	Top-1	Top-15	Top-16	Long	Short	
Perceived Latency ⇓	-3.1 s	-26 ms	77 ms	-3.2 s	-110 ms	213 ms
Error Rate ⇓	66 %	2.6%	0%	55 %	34 %	0%

Table 3: Average perceived latency and error rate for *Gesture Vocabulary's* gesture predictions. LSTM predictions 1 through 15 and the linear splines generate audio outputs that, on average, precede the completion of a gesture. Top-16 LSTM prediction (77 ms) falls within the *micro* scale.

Table 3 and Figure 5 show the trade-off between error rate and perceived latency with Spec-Ex. The LSTM and spline prediction methods generated audio before gesture completion and achieved lower error rates than randomly generated audio samples. However, the LSTM generated multiple audio samples, allowing it to outperform the long-term spline on all but its first prediction and to surpass the short-term spline 2 seconds before the gesture's completion (the Top-6 prediction, with a 33% error rate).

To synchronize with a user's gestures, the system must generate and play audio before the gesture is completed. The average predictions from Top-1 (66 % error rate) to Top-15 (2.6% error rate) were able to finish generation before gesture completion, allowing *Gesture Vocabulary* to **generate a correct audio output before a gesture is completed 97% of the time.** Comparatively, randomly selecting 15 outputs would result in a 6.25% error rate.

Because this prototype version of the system used a limited number of gesture options (Table 2), it was possible to generate audio to match all possible predictions within the limits of Roads' *micro* scale (1-100 ms): the Top-16 predictions of the LSTM resulted in an average perceptual latency of 77 ms. This shows that, even in the worst-case scenario (3% of the time), the speculative approach can complete its audio output more responsively than audio generation without Spec-Ex [7, 37]. This allows the system to begin playing audio based on an earlier guess at the time of gesture completion, then switch after 77 ms if that original guess is incorrect (the "rollback" described in Section 3.2).

As shown in Figure 5, the LSTM predictions most dramatically outperform the random choice with the earlier set of predictions. This, combined with the fact that the model generated most of its audio output before gesture completion, suggests opportunities for parameter optimization for the user or designer of a Spec-Ex-based interactive generative music system. For example, if the user wants to generate as few outputs as possible, they can adjust the prediction window. This prototype predicted gesture locations up to 3.6 seconds in the future, but higher accuracy may be possible with a shorter window.

6 Conclusion & Future Work

In this paper, we discuss the use of speculative execution (Spec-Ex) to reduce effective latency in a system for real-time gesture-controlled music generation. By predicting a user's gestures and generating audio in advance, a generative music system can enhance its real-time capabilities and suitability for live performance, allowing it to "shift" how the model's output is perceived across Curtis Roads' time scales. We present a prototype of *Gesture Vocabulary*, a system that uses Spec-Ex alongside gestural recognition to prompt a generative audio model with predicted inputs and synchronize its output with a user's gestures. Spec-Ex allows *Gesture Vocabulary* to generate audio before a gesture is completed and to "roll back," or correct its output within a time scale akin to instrument performance (1-100 ms) when it is incorrect. This allows the system to respond to a user with a perceptual latency much lower than the typical delay between large language model prompting and audio output (213 ms).

The prototype in this paper demonstrated the effectiveness of Spec-Ex, albeit with the limitations of a small number of gesture options. Generating the entire set of options was feasible; future evaluation will include more complex and varied gesture-to-text mappings, as needed for a predictive model to accurately generate audio before a gesture ends. Additionally, this prototype handled discrete gestures and mapped them to a discrete set of controls for the output generative audio model, a simplification that does not incorporate the physical affordances of speed and trajectory, which would add expressivity and complexity to how humans perceive each other's communicative gestures. Further development of *Gesture Vocabulary* will incorporate joint embeddings between language descriptions of gestures and musical properties [29] to enhance controllability and human understandability, while enabling transitions between known labels to account for the complexities and open-ended nature of physical motion.

7 Ethical Standards

This work was supported by NSF Award Number 2300633. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. *Gesture Vocabulary* uses the publicly available trained MediaPipe pose estimation model, Stable Audio Open Small for audio generation, and existing hand sign and motion classifications from public repositories. The only additional data used by the system is recorded and stored by the user in real time as they interact with the system, and is saved as a text file that represents gestures numerically rather than as images.

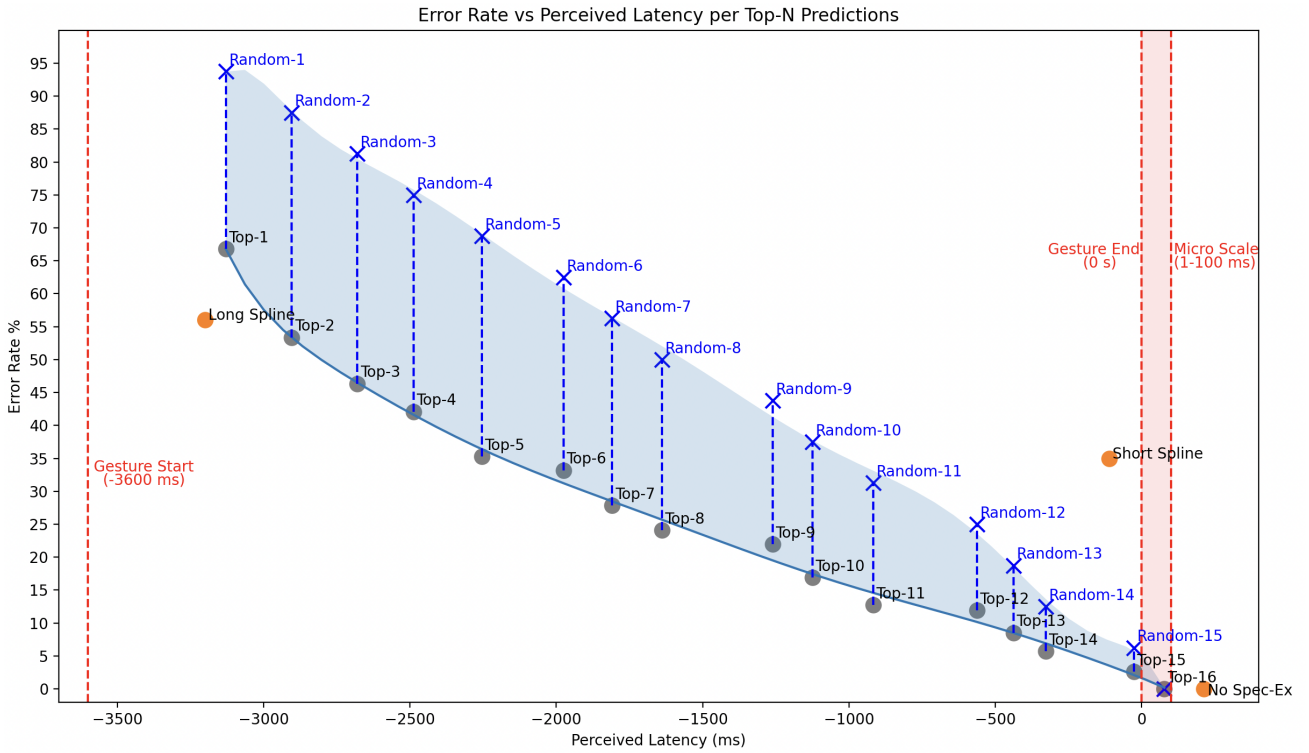

Figure 5: Error rate and average perceived latency for each Top-N prediction after 25 minutes of performance with *Gesture Vocabulary*. Top-1 (-3.1 s) through Top-15 (-27 ms) are able to generate audio before the completion of a gesture. Top-16 (77 ms), which is perfectly accurate, is able to generate audio within the *micro* time scale.

References

[1] Melek Alaftekin, Ishak Pacal, and Kenan Cicek. Real-time sign language recognition based on yolo algorithm. *Neural Computing and Applications*, 36(14):7609–7624, 2024.

[2] Misagh Azimi and Mo H Zareei. Live improvisation with fine-tuned generative ai: A musical metacreation approach. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 389–393, 2025.

[3] Zalán Borsos, Matt Sharifi, Damien Vincent, Eugene Kharitonov, Neil Zeghidour, and Marco Tagliasacchi. Soundstorm: Efficient parallel audio generation. *arXiv preprint arXiv:2305.09636*, 2023.

[4] Jonas Braasch. The telematic music system: Affordances for a new instrument to shape the music of tomorrow. *Contemporary Music Review*, 28(4-5):421–432, 2009.

[5] Juan-Pablo Cáceres and Chris Chafe. Jacktrip: Under the hood of an engine for network audio. *Journal of New Music Research*, 39(3):183–187, 2010.

[6] Antoine Caillon and Philippe Esling. Rave: A variational autoencoder for fast and high-quality neural audio synthesis. *arXiv preprint arXiv:2111.05011*, 2021.

[7] Antoine Caillon, Brian McWilliams, Cassie Tarakajian, Ian Simon, Ilaria Manco, Jesse Engel, Noah Constant, Pen Li, Timo I. Denk, Alberto Lalama, Andrea Agostinelli, Anna Huang, Ethan Manilow, George Brower, Hakan Erdogan, Heidi Lei, Itai Rolnick, Ivan Grishchenko, Manu Orsini, Matej Kastelic, Mauricio Zuluaga, Mauro Verzetti, Michael Dooley, Ondrej Skopek, Rafael Ferrer, Zalán Borsos, Äaron van den Oord, Douglas Eck, Eli Collins, Jason Baldridge, Tom Hume, Chris Donahue, Kehang Han, and Adam Roberts. Live music models. *arXiv:2508.04651*, 2025.

[8] Antonio Camurri, Barbara Mazzarino, Matteo Ricchetti, Renee Timmers, and Gualtiero Volpe. Multimodal analysis of expressive gesture in music and dance performances. In *International gesture workshop*, pages 20–39. Springer, 2003.

[9] Antonio Camurri, Gualtiero Volpe, Giovanni De Poli, and Marc Leman. Communicating expressiveness and affect in multimodal interactive systems. *Ieee Multimedia*, 12(1):43–53, 2005.

[10] Franco Caspe, Andrew McPherson, and Mark Sandler. Waveform autoencoding at the edge of perceivable latency. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 73–76, 2025.

[11] Franco Caspe, Jordie Shier, Mark Sandler, Charalampos Saitis, and Andrew McPherson. Designing neural synthesizers for low latency interaction. *arXiv preprint arXiv:2503.11562*, 2025.

[12] Chris Chafe, Juan-Pablo Caceres, and Michael Gurevich. Effect of temporal separation on synchronization in rhythmic performance. *Perception*, 39(7):982–992, 2010.

[13] Chris Chafe, Michael Gurevich, Grace Leslie, and Sean Tyan. Effect of time delay on ensemble accuracy. In *Proceedings of the international symposium on musical acoustics*, volume 31, page 46. Nara, 2004.

[14] Elaine Chew. About time: Stratedgies of performance revealed in graphs. *Visions of Research in Music Education*, 20(1):11, 2012.

[15] Elaine Chew, Alexander A Sawchuk, R Zimmerman, V Stoyanova, I Toshe, C Kyriakakis, C Papadopoulos, ARJ Franücois, and A Volk. Distributed immersive performance. *Annual Nat. Assoc. of the Schools of Music*, 2006.

[16] Jacob K Christopher, Brian R Bartoldson, Tal Ben-Nun, Michael Cardei, Bhavya Kailkhura, and Ferdinando Fioretto. Speculative diffusion decoding: Accelerating language generation through diffusion. *arXiv preprint arXiv:2408.05636*, 2024.

[17] Mark Cook. *Telematic music: History and development of the medium and current technologies related to performance*. PhD thesis, Bowling Green State University, 2015.

[18] Anton Ehlert. Improving input prediction in online fighting games, 2021.

[19] Sidney S Fels and Geoffrey E Hinton. Glove-talk: A neural network interface between a data-glove and a speech synthesizer. *IEEE Transactions on Neural Networks*, 4(1):2–8, 1993.

[20] Sidney S Fels and Geoffrey E Hinton. Glove-talk ii-a neural-network interface which maps gestures to parallel formant speech synthesizer controls. *IEEE Transactions on Neural Networks*, 8(5):977–984, 1997.

[21] Sidney S Fels, Bob Pritchard, and Allison Lenters. Fortouch: A wearable digital ventriloquized actor. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 274–275, 2009.

[22] Rebecca Fiebrink. Machine learning as meta-instrument: Human-machine partnerships shaping expressive instrumental creation. In *Musical instruments in the 21st century*, pages 137–151. Springer, 2017.

[23] Rebecca Fiebrink and Perry R Cook. The wekinator: a system for real-time, interactive machine learning in music. In *Proceedings of The Eleventh International Society for Music Information Retrieval Conference (ISMIR 2010)(Utrecht)*, volume 3, pages 2–1, 2010.

[24] Rebecca Fiebrink and Laetitia Sonami. Reflections on eight years of instrument creation with machine learning. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 237–242, 2020.

[25] H Flores_Garcia, P Seetharaman, R Kumar, and B Pardo. Vampnet: Music generation via masked acoustic token modeling. 24th International Society for Music Information Retrieval Conference, 2023.

[26] Óscar Fontenla-Romero, Bertha Guijarro-Berdiñas, David Martinez-Rego, Beatriz Pérez-Sánchez, and Diego Peteiro-Barral. Online machine learning. In *Efficiency and Scalability Methods for Computational Intellect*, pages 27–54. IGI Global, 2013.

[27] Ken Fujiwara, Rens Hoegen, Jonathan Gratch, and Norah E. Dunbar. Synchrony facilitates altruistic decision making for non-human avatars. *Computers in Human Behavior*, 128:107079, 2022.

[28] Hugo Flores García, Oriol Nieto, Justin Salamon, Bryan Pardo, and Prem Seetharaman. Sketch2sound: Controllable audio generation via time-varying signals and sonic imitations. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

[29] Qingqing Huang, Aren Jansen, Joonseok Lee, Ravi Ganti, Judith Yue Li, and Daniel PW Ellis. Mulan: A joint embedding of music audio and natural language. *arXiv preprint arXiv:2208.12415*, 2022.

[30] Robert Hupke, Dürre Jan, Norbert Werner, and Jürgen Peissig. Latency and quality-of-experience analysis of a networked music performance framework for realistic interaction. In *Audio Engineering Society Convention 152*. Audio Engineering Society, 2022.

[31] Robert H Jack, Tony Stockman, and Andrew McPherson. Effect of latency on performer interaction and subjective quality assessment of a digital musical instrument. In *Proceedings of the audio mostly 2016*, pages 116–123. 2016.

[32] David Kim-Boyle. Collaborative musical expression through interactive vr scores. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 126–132, 2025.

[33] Nelson Posse Lago and Fabio Kon. The quest for low latency. In *ICMC*, 2004.

[34] Zijian Lin, Yang Zhang, Yougen Yuan, Yuming Yan, Jinjiang Liu, Zhiyong Wu, Pengfei Hu, and Qun Yu. Accelerating autoregressive speech synthesis inference with speech speculative decoding. *arXiv preprint arXiv:2505.15380*, 2025.

[35] Alain Lioret, Lior Diler, Sami Dalil, and Marion Mota. Hybrid prediction for games' rollback netcode. In *ACM SIGGRAPH 2022 Posters*, pages 1–2. 2022.

[36] Mohammad YM Naser and Sylvia Bhattacharya. Empowering human-ai teams via intentional behavioral synchrony. *Frontiers in Neuroergonomics*, 4:1181827, 2023.

[37] Zachary Novack, Zach Evans, Zack Zukowski, Josiah Taylor, CJ Carr, Julian Parker, Adnan Al-Sinan, Gian Marco Iodice, Julian McAuley, Taylor Berg-Kirkpatrick, et al. Fast text-to-audio generation with adversarial post-training. *arXiv preprint arXiv:2505.08175*, 2025.

[38] Pauline Oliveros, Sarah Weaver, Mark Dresser, Jefferson Pitcher, Jonas Braasch, and Chris Chafe. Telematic music: six perspectives. *Leonardo Music Journal*, 19(1):95–96, 2009.

[39] Patrick O'Reilly, Julia Barnett, Hugo Fores Garcia, Annie Chu, Nathan Pruyne, Prem Seetharaman, and Bryan Pardo. The rhythm in anything: Audio-prompted drums generation with masked language modeling. 2025.

[40] Curtis Roads. The perception of microsound and its musical implications. *Annals of the New York Academy of Sciences*, 999(1):272–281, 2003.

[41] Curtis Roads. Rhythmic processes in electronic music. In *ICMC*, 2014.

[42] Jason Brent Smith and Jason Freeman. Adaptation and perceived creative autonomy in gesture-controlled interactive music. In *Proceedings of the 25th international conference on New Interfaces for Musical Expression*, 2025.

[43] Giuseppe Torre, Kristina Andersen, and Frank Baldé. The hands: The making of a digital musical instrument. *Computer Music Journal*, 40(2):22–34, 2016.

[44] Laurens Van Der Maaten, Eric O Postma, H Jaap Van Den Herik, et al. Dimensionality reduction: A comparative review. *Journal of machine learning research*, 10(66-71):13, 2009.

[45] Marcelo M Wanderley. Prehistoric nime: Revisiting research on new musical interfaces in the computer music community before nime. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 60–69, 2023.

[46] Scott S Wiltermuth and Chip Heath. Synchrony and cooperation. *Psychological science*, 20(1):1–5, 2009.

[47] Xin Zeng, Xiaoyu Wang, Tengxiang Zhang, Chun Yu, Shengdong Zhao, and Yiqiang Chen. Gesturegpt: Toward zero-shot free-form hand gesture understanding with large language model agents. *Proceedings of the ACM on Human-Computer Interaction*, 8(ISS):462–499, 2024.

[48] Chen Zhang, Zhuorui Liu, and Dawei Song. Beyond the speculative game: A survey of speculative execution in large language models. *arXiv preprint arXiv:2404.14897*, 2024.

[49] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*, 2020.