Cross-Modal Sig2Sig Machine Translation with Deep Generative Modeling for NIME Design

Lucy Strauss
lstra003@gold.ac.uk
Goldsmiths, University of London
London, United Kingdom

Prashanth Thattai Ravikumar
P.ThattaiRavikumar@gold.ac.uk
Goldsmiths, University of London
London, United Kingdom

Matthew Yee-King
m.yee-king@gold.ac.uk
Goldsmiths, University of London
London, United Kingdom

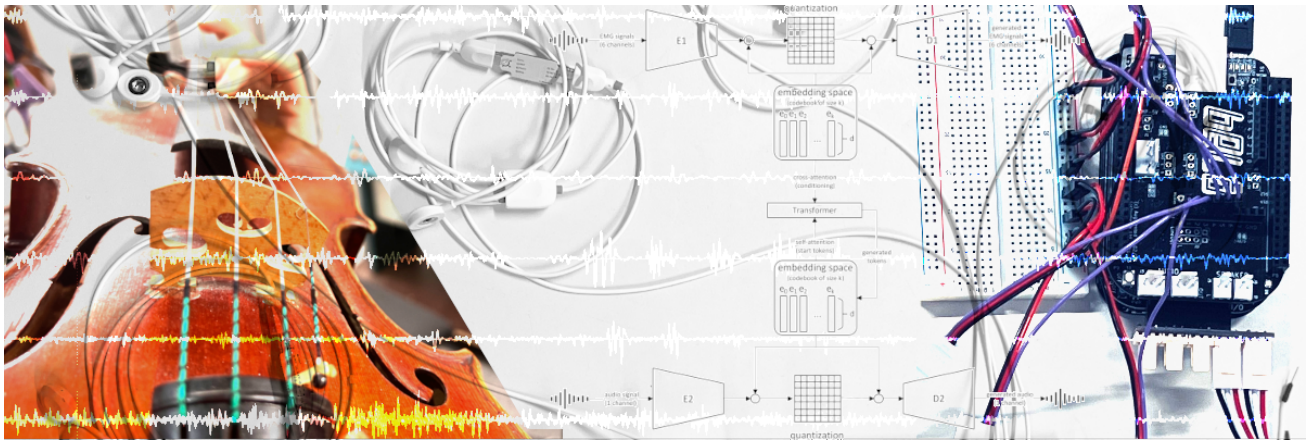


Figure 1: Composite image featuring superimposed elements: a viola photographed from the musician's perspective, electrode cables, EMG signal waveforms, a deep generative model network architecture diagram, a Bela board.

Abstract

NIME researchers frequently work with sensor signals that lack interpretability, such as signals from movement sensors and bioelectric sensors. However, there is a lack of NIME-specific approaches for building and evaluating deep generative models (DGM) of such signals, even though DGM are increasingly prevalent in NIME.

Our research focuses on cross-modal Sig2Sig machine translation, a sensor-sound mapping task using DGM. We present the Muscle-Listening Machine Learning Model for Live Music (MLMLMLM), a novel DGM intended for use within an interactive music system. MLMLMLM is trained on a bespoke time-aligned dataset of audio and electromyographic (EMG) signals and features a decoder-only Transformer and two RVQ-VAEs.

We position the technical work of designing bespoke DGM architectures as a NIME practice in its own right and employ a Technical Practice Research (TPR) approach to document the process of building MLMLMLM. Through our TPR process, a new evaluation method emerged for DGM with low-interpretability signals.

The contributions of this research are two-fold: 1) a novel DGM architecture for EMG-conditioned sequence generation of audio signals; 2) a method for more effectively developing and evaluating DGMs of multi-channel time-domain signals with low-interpretability.

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

Keywords

Deep Generative Modeling, cross-modal, Sig2Sig, Transformer, Variational Autoencoder, interactive music system, EMG, Technical Practice Research

1 Introduction

Mapping between sensor signals and sound is a strong theme in NIME since the inception of the conference. Recent advances in Deep Generative Modeling (DGM) have transformed cross-modal machine translation in other research areas. However, there are obstacles when attempting to transfer this interdisciplinary knowledge to NIME design.

In practice, custom DGMs undergo lengthy development processes before culminating in functioning, usable arefacts. Fellow NIME researchers have proposed innovative technical and methodological solutions to this problem.

Technical solutions include specialized pipelines that expedite functioning prototypes [27] [29]. The nature of DGM development also gives rise to methodological implications for NIME design; much technical design work of DGM architectures is overlooked by traditional human-computer interaction methodologies.

To close this methodological gap, Technical Practice Research (TPR) [28] positions technical work as a practice in an of itself. Pelinski et al. [28] describe TPR as:

> "an alternative mode of research in technical practice that places the locus of knowledge production in the practice rather than the technical artefact, by focusing on the first-person and real-time nature of technical practice." [28]

Complementary to technical and methodological solutions, we propose building specialized knowledge of DGM within NIME to streamline knowledge transfer from other DGM literature.

Such NIME-specific knowledge can include novel methods for designing custom DGM architectures for computing tasks at the heart of NIME research. Our research broadly encompasses the task of cross-modal mapping between signal data–or more specific to DGM, *cross-modal Sig2Sig machine translation*.

We present the TPR process of designing the Muscle Listening Machine Learning Model for Live Music (MLMLMLM), a DGM that generates sequences of audio signals conditioned on electromyographic (EMG) signals from a performing musician.[1] MLMLMLM is intended for use within an interactive music system for live performance settings.[2]

MLMLMLM is trained on a bespoke dataset of time-aligned audio and EMG signals of viola playing. The MLMLMLM architecture comprises two Residual Vector-quantized Variational Autoencoders (RVQ-VAE) and a cross-attention Transformer decoder in latent space. The RVQ-VAEs each perform representation learning and latent space mapping of audio and EMG signals respectively. The Transformer performs causal, autoregressive sequence generation with streaming conditioning.[3]

This paper makes two main contributions. The first is a method contribution that emerged from the TPR process: a novel evaluation method for use while developing DGMs of multi-channel signals with low-interpretability. The second contribution is an artefact: the MLMLMLM architecture itself.

We provide a background of machine translation, relevant DGM architectural classes, and technical challenges (Section 2). Section 3 presents our novel method for building DGMs. Section 4 presents the experiment design. In Sections 5, 6 & 7, we provide a TPR account of the MLMLMLM development process. We include a discussion (Section 8), and final remarks.

2 Background

The RAVE architecture [5] is well-represented in the literature concerning audio DGM within interactive music systems. In the interest of advancing NIME research, Caspe et. al [6] make a strong case for researchers to redesign and iterate upon existing architectures, instead of considering existing models such as "RAVE as a one-size-fits-all tool" [6].

Likewise, Jourdan and Caramiaux [17] call attention to the limited number of DGM techniques and rarity of custom model architectures in NIME, for which a possible influencing factor is specialized technical knowledge required to build custom architectures [17].

These insights point towards a need for NIME-compatible methods for DGM development. Moreover, NIME is not only concerned with audio; we also work with movement sensors. We therefore frame this need in more NIME-specific terms as a cross-modal mapping problem.

In technoscientific literature, *machine translation* is a multi-faceted computing task with a long history, stretching back to the mid-20th century [30] [42]. Poibeau's foundational book on the subject defines machine translation tools as "computer programs capable of automatically producing in a target language the translation of a text in a source language" [30]. The book portrays translation as a complex concept that is not simple to define, and thus open to interpretation. Furthermore, Poibeau emphasizes that there are many different processes involved in translation [30].

Machine translation traditionally entails working with text and language, but also holds relevance to other domains. Translation is also possible between data modalities. For example, Bisig and Tatar translate pose sequences into raw audio waveforms [3]. *Cross-modal* machine translation gives rise to unique challenges stemming from mismatches in internal temporo-structural organization, as well as inherent cross-modal differences in dimensionality [39] [18]. Cross-modal machine translation frequently involves continuous data domains.[4] Specifically, Sig2Sig is a translation task that models relationships between signals by learning a mapping between two signal domains [20].

In the paragraphs that follow, we provide an overview of DGM architectures most relevant to *cross-modal Sig2Sig machine translation* to sharpen our scope to the NIME-specific problem of cross-modal mapping.

The multi-faceted characteristic of machine translation is evident from the prominence of composed state-of-the-art model architectures. In the generative music context, Stable Audio features a Diffusion Transformer in combination with other architectural elements, including a VAE that models raw audio signals[5] [13] [14]. The Diffusion Transformer translates between text embeddings and audio embeddings in the VAE latent space. The strategy of composing model architectures from different models is typical of machine translation architectures, possibly because translation inherently involves multiple tasks, as articulated by Poibeau [30].

Transformers [38] are prevalent in state-of-the-art machine translation architectures, having lead to rapid advancements in machine translation in recent years [42]. Transformers are characterized by attention mechanisms, as opposed to recurrence or convolution and have undergone significant developments since their invention, such as cross-attention conditioning through dual-branch architectures [7].

Transformers have been used in speech-to-speech translation architectures using mel-spectrogram representations of audio signals [19]. Transformers also perform well for DGM of musical audio signals [41]. However, Transformers featuring in previous NIME proceedings only model MIDI representations of music [23] [16] [12] [25].

DGMs capable of generating bio-electric signals are unusual, where bio-electric signals are usually framed as control signals in regression or classification tasks. Mainolfi translates audio to electroencephalographic signals [24], the opposite of our task to translate bio-electric signals to audio. More closely related to our research, the CAVI system generates predicted EMG signals [10].

A challenge of DGM of non-audio signal data is a lack of interpretability. Whereas humans can easily make quick judgments about musical audio signal quality by listening, we are less perceptually attuned to non-musical, non-audio signals. NIME researchers utilize control signals from a varied range of sensor modalities, such as EMG signals [31] [8] [22] [40] [9] [26] [11], yet EMG signals notoriously lack interpretability [2]. Our research lays groundwork for DGM of low-interpretability signals in NIME.

[1] The MLMLMLM codebase is available in our open-source GitHub repository: https://github.com/lucystrauss/MLMLMLM

[2] Video of the first author performing live with MLMLMLM: https://www.youtube.com/watch?v=Y5qrDAH5ny0&list=PLrf6aP0TdLZeNa_MUND0_8IYpRLV-qyAz

[3] i.e. The model does not require the full conditioning sequence to be observed upfront and can thus can start generating outputs before the full conditioning signal is available–essential for interactive live performance settings.

[4] in contrast to discrete data such as text.

[5] as opposed to spectrogram representations of audio

3 Method

We present a novel method for building DGMs of multi-channel signals with low-interpretability. This method contribution emerged through our TPR process documented in Sections 5, 6 & 7.

In the practice of building and evaluating DGMs, researchers frequently inspect reconstructions and compare them with samples from the original dataset. These qualitative inspections enable researchers to make quick judgments about the generative capabilities of models-in-development, thus informing key decisions towards final model architectures and training hyperparameters.

Both aural and visual feedback are essential during audio DGM training because spectrograms and waveforms encode fundamentally different information about signals. Magnitude spectrograms depict useful information about frequency-domain structure and energy-based perceptual features of sound. However, spectrograms do not depict errors in audio fidelity that can be easily detected by listening to waveforms. As such, looking and listening serve crucial, complementary functions when modeling musical audio signals.

However, the method of qualitative aural and visual inspection is less informative when developing DGMs of non-audio signals that are less perceptually interpretable. Humans are perceptually attuned to notice variations in musical audio signal quality, yet we are not as perceptually well-equipped to make quick judgments about the fidelity of raw sensor signals without additional data-sound mapping designs [15]. We encountered this imbalance in our perceptual understanding between music and EMG signals during our TPR process, where EMG signals are less interpretable than audio signals.

To overcome the challenge of modeling less-interpretable signals, we devised a method whereby we first train a multi-modal DGM on a time-aligned dataset wherein one of the channels is audio and the remaining channels are EMG signals. This interim training phase facilitates a quick evaluation of generated signals through looking at and listening to interpretable audio channel reconstructions. Once the desired network architecture and training hyper-parameters are discovered, we remove the audio channel from the training dataset and train the model again. Through this process, we are able evaluate a multi-channel DGM of EMG signals.

Our method is not entirely peculiar in DGM practice. For example, Tatar et al. discover model hyperparameters by training on a reliable image dataset during bespoke audio DGM development [36].

We summarize our method contribution below:

(1) source (or create) a time-aligned, multi-modal dataset of which one modality is audio
(2) train model on signal modality with higher interpretability (eg. audio)
    - does the training script run?
    - can the network architecture model signal data?
    - is it necessary to make adjustments to training hyperparameters or loss balance?
(3) train model on samples of paired multi-modal signals
    - what adjustments are necessary to accommodate for the increased channel count?
    - adjust model architecture to discover model capacity requirements (eg. increase network width or depth)
    - tune training hyperparameters and loss function balance
(4) train model on less-interpretable, multi-channel signal data (without the audio channel) using the discovered architecture and hyperparameters

4 Experiment Design

The experiment design serves our design goal of a DGM capable of cross-modal Sig2Sig machine translation, wherein a DGM translates between raw signal data of two (or more) data modalities. Specifically, we sought to design a model that could translate EMG signals into viola-like audio signals.

We conceptualise machine translation as a multi-task process, as portrayed by Poibeau [30]. As such, we compartmentalize cross-modal Sig2Sig machine translation into three sub-tasks:

- Deep Generative Modeling of Audio Signals
- Deep generative modeling of EMG signals
- Cross-Modal Mapping & Latent Sequence Generation[6]

In this section, we present our dataset (4.1), the MLMLMLM architecture (4.2), and our TPR experiment approach (4.3).

4.1 Dataset

MLMLMLM is trained on a custom dataset of time-aligned EMG and audio signals of improvised viola playing. The dataset has 7 channels, including 1 audio channel and 6 EMG channels. We captured 130 minutes of multi-channel signal data and augmented the dataset using time-stretch operations, totaling 420 minutes of training data. EMG signal pre-processing included a 1Hz high-pass filter for offset removal and a 50Hz notch filter to suppress power-line interference.

Viola sound was captured at 44100Hz with a DPA violin/viola clip on condenser microphone for consistency with the intended real-world stage performance conditions at inference time.

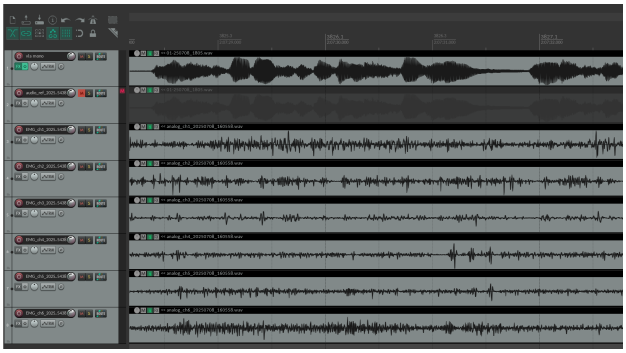
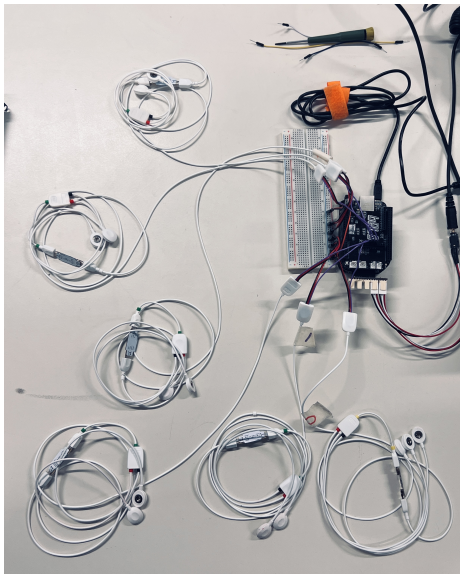


Figure 2: Screenshot showing the time-aligned multi-channel dataset in REAPER. From top to bottom, the tracks are: the high-quality viola audio track, the muted reference audio track, then the 6 EMG channels.

We recorded EMG data at 22050Hz on a Bela board with Plux BITalino EMG sensors, employing a non-invasive EMG sensing set-up with gelled surface electrodes.

Electrode placements were informed by our prior work, including VAE training on a dataset of EMG signals of viola playing [34], and the analysis of a specialized movement technique in viola performance that engages the right side of the violist's torso to produce subtle timbral variations on the instrument [33].

[6]We use a Transformer decoder architecture to perform both cross-modal mapping and latent sequence generation. We group both tasks into one sub-task point (rather than two points) to represent how we compartmentalized the overall design objectives in practice.



(a) Bela board and electrode cables used to capture EMG signals and the reference audio track. Note that the reference audio microphone is out-of-frame in this image.

(b) Violist's point of view during recording. The DPA clip-on microphone attached to the viola records high-quality audio, whereas the reference microphone is on the microphone stand in front of the musician. Also visible is a stereo-pair of microphones. We chose to model mono audio signals and therefore we did not use the audio from this stereo pair in the final dataset.

Figure 3: Images photographed at the dataset recording sessions.

In the current sensing set-up, one electrode is placed over the latissimus dorsi on the right side of the body to capture EMG information relating to the above-mentioned specialized viola playing technique. Other electrodes are placed over the forearms to capture wrist flexion; under the left forearm to capture wrist extension; the right deltoid to capture arm abduction; and the left bicep to capture vibrato and elbow flexion.

We recorded a reference audio channel through the Bela board using a dynamic microphone, and performed manual alignment in REAPER (Figure 2). We augmented the dataset using time-stretch, muted the reference audio track, and exported six-second long windows of the 7 channels for our training dataset as a multichannel waveform file.[7]

4.2 MLMLMLM Architecture

MLMLMLM performs cross-modal Sig2Sig machine translation, a computational task that we conceptualize and define herein. The architecture is composed of two RVQ-VAEs and a decoder-only Transformer. Each model is trained separately, to perform a sub-task of the multi-task machine-translation objective. One RVQ-VAE models EMG signals; the other models audio signals. The Transformer is implemented in latent space, using quantized latent vectors of the audio RVQ-VAE for self-attention, and quantized latent vectors of the EMG RVQ-VAE for cross-attention.

The EMG encoder produces a compressed representation informed by machine-derived feature extraction and can therefore be considered an information retrieval task. This latent signal representation is thus richer and more informative than raw signals, and of compressed dimensionality–a preferable conditioning signal, also in terms of computational efficiency.

We do not decode EMG signals at inference time. Nonetheless, we include EMG reconstruction loss during training, to encourage information preservation and compression of essential features without losing meaningful details. Although viola audio signals and EMG signals differ in frequency range, they share time- and frequency-domain characteristics. Consequently, many signal processing techniques are found in both audio and EMG signal processing literature, such as STFT for feature extraction and analysis [32] [35]. Thus, STFT is an appropriate choice for reconstruction loss of EMG signals, encouraging the network (including the EMG encoder) to extract meaningful features.

Our trained MLMLMLM model generates raw audio signal sequences up to 6 seconds in length, a limitation resulting from hardware constraints.[8] We find this a reasonable length for musical phrases because MLMLMLM is intended for interactive performance settings with continuous input from the human musician, who can influence structure by providing new audio

[7] The .lua script for exporting overlapping windows of multichannel wave files from REAPER is available in the MLMLMLM GitHub repository.

[8] Memory requirements for Transformers generally scale quadratically with sequence length. VRAM hardware constraints are relevant during training, as well as during inference time. For example, we currently have access to a laptop with NVIDIA 3070 GPU with 8GB VRAM for running MLMLMLM in live performance scenarios. With this set-up in mind, we did not train on longer sequences that would have exceeded the memory constraints of our available hardware for performances.

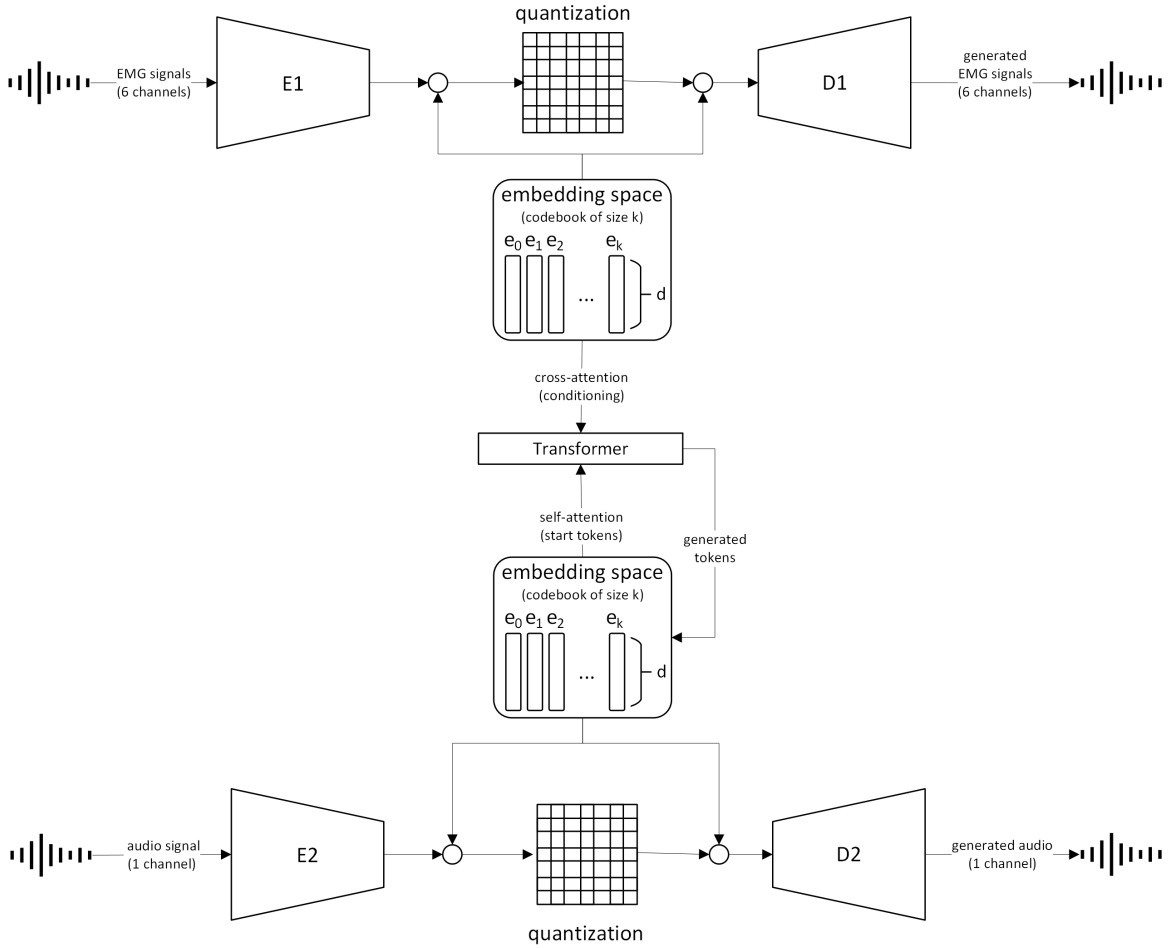


Figure 4: This diagram depicts the three main architectural components of MLMLMLM: a VAE of EMG signals (top); a VAE of audio signals (bottom); and a Transformer (middle) that translates between the VAE latent spaces with a cross-attention mechanism.

start tokens every few seconds.[9] However, the limit of 6 seconds can be extrapolated in future with specialized techniques [44].

4.2.1 RVQ-VAE Architecture. The main architectural components of the two VQ-VAEs are identical (Figure 5). We selected the Oobleck VAE (with snake activations) because it performs well in DGM of raw audio waveforms [14] [13]. Snake activations facilitate the reconstruction of signal phase information [45].

The MLMLMLM VAE bottlenecks are discrete to streamline conversion of latent vectors into token embeddings for the Transformer decoder. We selected RVQ bottlenecks [21] because they are well-documented[10] with robust open-source PyTorch implementations.[11]

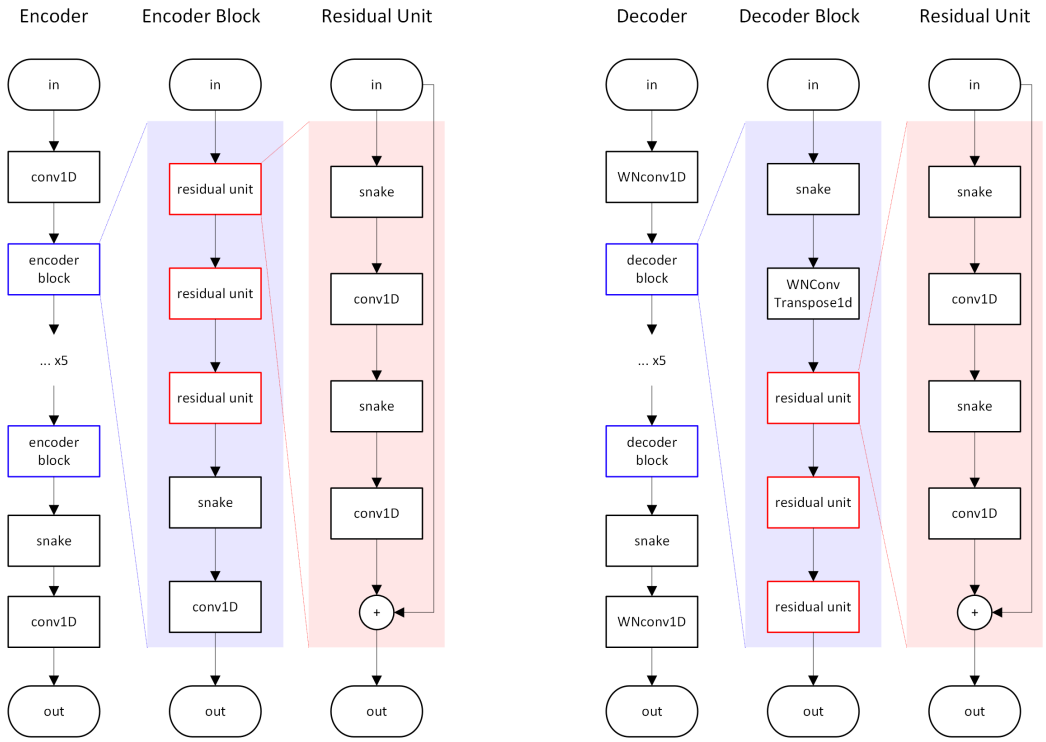
4.2.2 Cross-Attention Transformer. Similarly to MusicLM [1] and AudioLM [4], we use a decoder-only Transformer to focus the computing task on auto-regressive sequence generation. Furthermore, we chose Transformers over other neural audio synthesis approaches because our computing task includes cross-modal mapping. Transformers offer a solution for both cross-modal mapping & latent sequence generation with cross-attention and self-attention mechanisms respectively.

We selected the ContinuousTransformer architecture from the open-source stable-audio-tools codebase[12] for its robust PyTorch implementation. We modified the ContinuousTransformer model and implemented custom training scripts towards streaming conditioning during inference time. We included a cross-attention layer in the MLMLMLM Transformer decoder to facilitate a learned bridge between EMG and audio latent spaces.

The Transformer (Figure 6) operates in latent space dimensionality of the audio RVQ-VAE and the EMG RVQ-VAE.

4.2.3 Inference Pipeline. At inference time, we encode a short window of audio with the audio RVQ-VAE to produce a start token. The length of this audio (number of start tokens) is adjustable and the minimum length is 2048 samples. Live incoming EMG signals are encoded by the EMG RVQ-VAE, to acquire EMG latents. The Transformer generates future tokens, continuing

[9] A full, detailed description of the performance set-up is out of scope for this paper, but we include a supplementary video file that features a live performance with MLMLMLM, where the trigger mechanism for the model to start generating is a change above a user-defined threshold of the RMS moving average of the incoming acoustic instrument audio signal. Thereby, the musician decides when to prompt the model to 'listen' to live EMG signals an generate audio output. In this way, we use the sequence length limitation as a meta-compositional feature of the performance.
[10] https://drscotthawley.github.io/blog/posts/2023-06-12-RVQ.html
[11] https://github.com/lucidrains/vector-quantize-pytorch
[12] https://github.com/Stability-AI/stable-audio-tools



(a) The encoder of the RVQ-VAE architecture. This figure depicts the convolutional layers and activation layers for the encoder, encoder blocks, and residual layers.

(b) The decoder of the RVQ-VAE architecture. This figure depicts the convolutional layers and activation layers for the decoder, decoder blocks, and residual layers.

Figure 5: RVQ-VAE encoder and decoder diagrams shown side-by-side for comparison. Notice that the downsampling convolutional layer follows the residual units in the encoder, whereas the upsampling convolutional layer precedes the residual units in the decoder.

the latent audio sequence from the start tokens, while using incoming latent EMG tokens for cross-attention conditioning. The tokens generated by the Transformer are decoded using the audio RVQ-VAE.[13]

The Transformer is causal,[14] autoregressive, and facilitates token-by-token generation with streaming conditioning. These features enable us to start generating audio before the complete EMG conditioning sequence has been received, i.e. the model can receive a live incoming EMG conditioning stream and output generated audio simultaneously.[15] This updated functionality is essential for live interactive performance settings.

4.3 TPR Experiment Approach

We document our TPR process of building the MLMLMLM model architecture in the sections that follow. TPR was first proposed by NIME researchers, with the NIME context in mind [28], providing a means to share insights that would traditionally be excluded from technical reports and user studies. The need for such a methodology is evidenced by the current lack of bespoke DGM architectures in the NIME literature. In the spirit of TPR, we embrace the non-linear nature of this process and visibilize decision-making towards a nuanced vocabulary for thinking about model training in the NIME context. We consider detailed accounts of process a strength of the TPR approach.

With this view, we document how frequent evaluations of our model-in-development informed the decisions towards the current MLMLMLM architecture. We developed MLMLMLM in stages, focusing on different machine translation sub-tasks at different moments during the design process. These tasks include:

- Phase 1: Deep Generative Modeling of Audio Signals (Section 5)
- Phase 2: Deep generative modeling of EMG signals (Section 6)
- Phase 3: Cross-Modal Mapping & Latent Sequence Generation (Section 7)

In the sections that follow, we recount our TPR process for each phase. We tabulate details from training runs for ease of comparison and share insights that we gained by observing original and reconstructed signals, for both audio and EMG signals. Our primary criterion for judging the success of experimental training runs was the variety of model outputs, given that model needs to be responsive to a range of metrics in a live performance intended use case. It is through this TPR process that our

[13] At inference time, we do not use the EMG RVQ-VAE decoder; only need the encoder is needed to produce latent vectors of encoded EMG signals.
[14] Both the generation modality (self-attention) and conditioning modality (cross-attention) are causal.
[15] This I/O streaming involves asynchronous programming. We recommend using the pybela library https://github.com/BelaPlatform/pybela/tree/main/pybela.

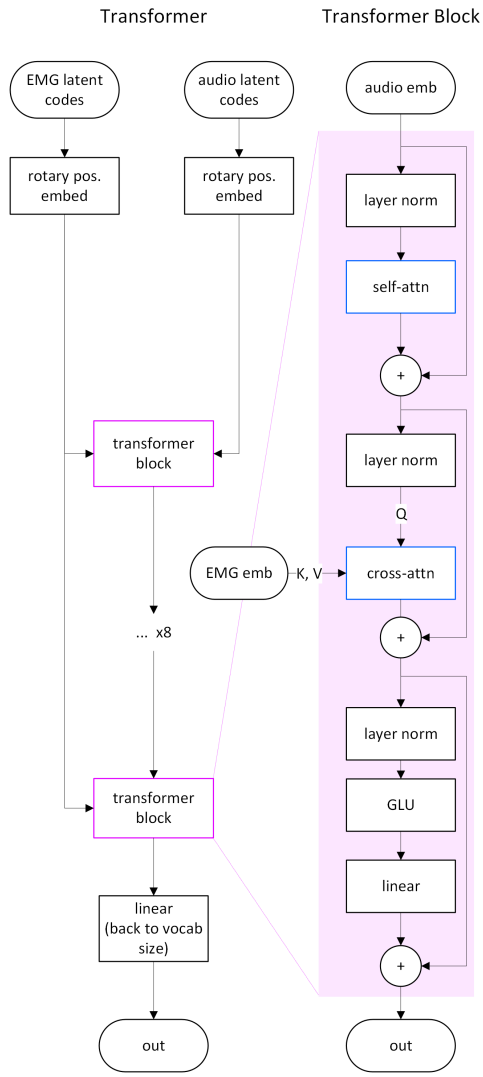


Figure 6: The decoder-only Transformer architecture in MLMLMLM. The Transformer is on the left and the Transformer Block is on the right.

method contribution (Section 3) and MLMLMLM architecture contribution (Section 4.2) emerged.

Reconstructions from a selection of experiments are presented visually in Appendix A and B.[16]

5 Phase 1: Deep Generative Modeling of Audio Signals

During these early stages of technical practice, we were still undecided on the computing method that we would use for cross-modal translation. Thus, our point of entry was to train a βVAE for signal DGM.

The models in the following experiments are trained on one channel of raw audio waveforms from our custom dataset (Section 4.1). Our primary objective at this stage was to train the VAE architecture from scratch on custom dataset audio. Thus, Phase 1 primarily investigates whether our dataset size is sufficient to train this particular VAE architecture.

5.1 Experiments

Table 1 presents architectural details and training hyperparameters; Table 2 presents results. For brevity, we use 'A' to indicate that the input and generation modality is 1 channel of audio. RVQ-VAE experiments are numbered 1 - 5 (eg. 'A RVQ-VAE 1', and so forth).

First, we trained a βVAE using the hyperparameters recommended for the Oobleck VAE architecture [14]. We name this experiment Audio VAE (A VAE). We observed that the model reconstructs pitch and loudness, but reconstructed audio sounds tinny compared to the original audio. Having worked with Stable Audio models before,[17] we were aware of this sound quality limitation of the model architecture beforehand, and deemed training experiment A VAE successful–our dataset size was sufficient.

We then considered our ultimate research goals of cross-modal Sig2Sig machine translation, identifying cross-attention Transformers as a promising technical solution. In preparation for pursuing this approach, we exchanged the standard VAE bottleneck for a vector-quantized latent space, towards modeling sequences of discrete latents with a Transformer at a later stage. The reconstruction loss remained a Multi-Resolution STFT (MRSTFT) calculation. We omitted the explicit KLD term and replaced it with a quantizer loss, comprising a codebook loss and a commitment loss averaged over 4 residual quantizers. The quantizer loss is implemented within the ResidualVQ module of the `vector_quantize_pytorch` library,[18] based on [43]. It is standard practice to omit KLD in the VQ-VAE literature; "Since we assume a uniform prior for z, the KL term that usually appears in the ELBO is constant w.r.t. the encoder parameters and can thus be ignored for training" [37].

For A RVQ-VAE 1, we used the same architecture and training dynamics as A VAE, but halved the sequence length.[19] Listening back to A RVQ-VAE 1 reconstructions, we observed a similar audio quality to the A VAE reconstructions. However, we could hear subtle audio artifacts at regular intervals–possibly a result of vector quantization.

With A RVQ-VAE 2, we investigated whether a larger latent dimensionality could eliminate regular audio artifacts. After 86 epochs, MRSTFT loss was greater and we could not perceive a reduction in regular audio artifacts from the quantizer. Although the feature-matching and discriminator loss were smaller, these improvements were not sufficient to justify the larger latent dimensionality at the expense of computational and memory efficiency for the Transformer at a later stage. Although A RVQ-VAE 2 was not unsuccessful in learning from our dataset, we stopped A RVQ-VAE 2 early at 86 epochs.

For A RVQ-VAE 3, we reverted back to original architectural settings to explore training on a longer sequence length, omitting the discriminator.[20] We found that quantizer and MRSTFT losses

[16] A verbose selection of audio and visual training feedback is available online: https://lucystrauss.github.io/NIME_2026_examples/

[17] https://lucystrauss.com/tech-tea-exchange

[18] https://github.com/lucidrains/vector-quantize-pytorch/tree/a8235c854f9abdf8f4ba5d4907be25c045d3c4af.

[19] The additional architectural elements required for the quantized latent space required greater VRAM (GPU memory). We compromised on sequence length to avoid a very small batch size (very small batch sizes can lead to noisy gradients and destabilize training).

[20] We omitted the discriminator due to hardware memory constraints. This enabled us to speed up training by using a larger sequence length, thereby saving on computing resources.

Table 1: Comparison of architectural and training settings experimental training runs for DGM of one audio channel. Wherever a value is —, it is because that metric was not calculated for that training run.

	dataset size	num conv. channels	num latent dimensions	KLD β	learning rate	sequence length	codebook size
A VAE	unaugmented	128	64	1×10^{-4}	1.5×10^{-4}	65536	—
A RVQ-VAE 1	augmented	128	64	—	1.5×10^{-4}	32768	2048
A RVQ-VAE 2	augmented	128	128	—	1.5×10^{-4}	32768	2048
A RVQ-VAE 3	augmented	128	64	—	1.5×10^{-4}	196608	2048

Table 2: Comparison of results of experimental training runs for DGM of one audio channel. The 'A RVQ-VAE 1' run is depicted twice for the respective purposes of: 1) presenting loss values at epoch 86 for ease of comparison with the 'A RVQ-VAE 2' run; 2) presenting final loss values at epoch 186. All labeled columns are losses averaged over training steps within the epoch, except 'epoch' and 'inspection'. The 'inspection' column summarizes the qualitative judgments we made during audio and visual inspection of generated modal outputs. Wherever a value is —, it is because that metric was not calculated for that training run.

	epoch	MRSTFT↓	KL loss↓	quantizer↓	adversarial↓	feature-matching↓	disc.↓	train↓	inspection
A VAE	186	0.905587	**0.025764**	—	0.080731	0.065305	1.565698	1.077424	successful
A RVQ-VAE 1	86	0.881584	—	0.296440	0.055323	0.065031	1.574482	1.298557	successful
A RVQ-VAE 1	186	0.832627	—	0.358071	**0.033008**	0.070342	1.801470	1.294094	successful
A RVQ-VAE 2	86	0.882269	—	0.291433	0.046758	**0.061129**	**1.554498**	1.281801	successful
A RVQ-VAE 3	186	**0.620433**	—	**0.152697**	—	—	—	**0.773132**	successful

were greatly improved by these adjustments[21] and reconstructions sounded–to our ears–very similar to those from previous runs.

6 Phase 2: Deep Generative Modeling of EMG Signals

The heart of our method contribution lies in this section. The Oobleck VAE of Stable Audio was intended to generate 1 or 2 channels [13]. Here, we document our process of adapting the working audio DGM to account for the increased data-space dimensionality of 6 EMG signal channels.

In practice, working with EMG and audio signals is similar. Both are time-domain signals, thus many signal processing and feature extraction techniques can be applied to both. Given these similarities, we chose to use the same audio VAE that we had already successfully trained (5.1) as a starting point to model EMG signals.

However, we suspected that we would need to adjust the model architecture and hyperparameters to account for the larger channel count. Our goal at this stage was to discover the appropriate adjustments of these interdependent components.

We entered Phase 2 knowing that EMG signals are less interpretable than viola audio signals in terms of human perception. As such, we sought a solution whereby we could observe reconstructions output by the EMG model and quickly judge the reconstruction fidelity and accuracy in practice, without relying on quantitative metrics alone.

Our solution to the challenge of signal interpretability was to run a series of experiments modeling both modalities with a VAE (7 channels), then to use the discovered architecture and hyperparameters to train a 6-channel EMG VAE. Through this method, we discovered the appropriate architectural capacity and hyperparameters for the increased dataset channel count.

6.1 EMG+audio Experiments

We conducted the following training experiments where the VAE input and output was paired samples of EMG and audio data (6 channels of EMG signals and 1 channel of audio) totaling 7 channels. The objective was to discover settings for model architecture and training hyperparameters for the increased dataset channel count.

The loss was a combination of KLD and a combined MRSTFT reconstruction loss of all 7 channels. We used a KLD β of 1×10^{-4} as a starting point because this is the value stated in the original Stable Audio publication [13]. For all experiments, the batch size was 4 and the dataset was our custom dataset.

Architectural details and training hyperparameters from DGM of audio (1 channel) and EMG signals (6 channels) are tabulated in Table 3. Results are tabulated in Table 4. We refer to each training run as EA (EMG+audio), numbered 1 - 5 (eg. 'EA VAE 1', and so forth).

First, we investigated latent dimensionality. For EA VAE 1, we used 128 convolutional channels and a latent dimensionality of 128. We observed (Appendix A) a near-complete collapse of diversity in reconstructions,[22] and therefore deemed EA VAE 1 unsuccessful.

For EA VAE 2, we explored whether a latent dimensionality of 128 was indeed too large, even when the number of convolutional parameters is increased to 196 to account for the additional dataset channels. Our observations were the same as for EA VAE 1.

Given that increased latent dimensionality did not prevent collapse in effective generative diversity, we maintained the larger

[21]The improvements to MRSTFT and quantizer losses are unsurprising because the discriminator would have introduced a competing training objective.

[22]i.e. All generated outputs are visually and audibly indistinguishable, suggesting convergence to a single average-like solution.

Table 3: Comparison of architectural and training settings experimental training runs for DGM of 7 signal channels in total (comprising 1 audio channel and 6 EMG channels). Wherever a value is —, it is because that metric was not calculated for that training run.

	dataset size	num conv channels	num latent dimensions	KLD β	learning rate	sequence length
EA VAE 1	augmented	128	128	1×10^{-4}	1.5×10^{-4}	32768
EA VAE 2	augmented	196	128	1×10^{-4}	1.5×10^{-4}	32768
EA VAE 3	augmented	196	64	1×10^{-5}	1.5×10^{-5}	32768
EA VAE 4	augmented	196	64	1×10^{-4}	1.5×10^{-4}	32768

Table 4: Comparison of results of experimental training runs for DGM of 7 signal channels in total (comprising 1 audio channel and 6 EMG channels). All runs used standard VAE architecture (i.e. *without* a quantized latent space). Therefore, quantizer loss is neither calculated nor depicted, unlike Tables 1 and 5. All labeled columns are losses averaged over logged training steps within the epoch, except 'epoch' and 'inspection'. The 'EA3' run is depicted twice for the respective purposes of: 1) presenting loss values at epoch 96 for ease of comparison with the 'EA2' run; 2) presenting final loss values at epoch 113. The 'inspection' column summarizes the qualitative judgments we made during audio and visual inspection of generated modal outputs.

	epoch	MRSTFT↓	KL loss↓	adversarial↓	feature-matching↓	disc.↓	train↓	inspection
EA VAE 1	113	1.406177	0.016579	0.065438	0.088408	1.559385	1.576601	unsuccessful
EA VAE 2	96	1.411290	0.000352	0.078537	0.120403	**1.445802**	1.610581	unsuccessful
EA VAE 3	96	1.099944	0.003866	**0.011029**	0.039306	1.804903	1.154145	successful
EA VAE 3	113	**1.085277**	0.003776	0.017864	**0.033965**	1.790194	**1.140882**	successful
EA VAE 4	113	1.372990	**0.000143**	0.056736	0.059722	1.685943	1.489592	unsuccessful

encoder and decoder capacity (196 convolutional channels), but reverted to a latent dimensionality of 64 for EA VAE 3.

Promisingly, we observed accurate, diverse reconstructions (Appendix B). The reconstructed EMG spectrograms seemed to reproduce the general shape of the original spectrograms. However, the spectrograms (both original and reconstruction) look noisy to the naked eye. Furthermore, we observe a loss of detail in the reconstructions that is especially noticeable above 35 Hz. We had anticipated some difference in quality between original and generated signals, but were unsure whether the loss of detail rendered our model unusable.

The additional feedback of audio reconstructions was helpful in this regard. We observed that the model was able to reconstruct exact audio pitches. By observing great improvements in audio signal modeling–improved reconstruction diversity–, we were able to make a quick judgment about the network's ability to model signals: that the EA3 network architecture is capable of modeling our multi-channel, multi-modal dataset.

Although we had already found an acceptable model architecture with usable training dynamics, we sought to understand how sensitive this architecture was to KLD β and learning rate. For EA VAE 4, we again observed collapse in effective generative diversity.

We were able to solve the problem of collapsed reconstruction diversity by increasing model capacity and lowering the KL weight. This indicates that the model was over-regularized relative to its representational capacity and that the failure arose from constrained information flow, not from adversarial instability or intrinsic mode collapse caused by the discriminator.

6.2 EMG Experiments

We ran further experiments with only the 6 EMG channels from our custom dataset. Table 5 presents architectural details and training hyperparameters. Table 6 presents results.

For EMG VAE, we first replicated experiment EA VAE 3–this time with only 6 EMG channels–to confirm whether the previously successful training dynamics and architectural settings would produce favourable results with an EMG-only dataset. Even with the absence of interpretable audio reconstructions, our visual observations were similar enough to our observations from EA VAE 3 that we supposed the model training was successful.

We then added vector quantization for EMG RVQ-VAE (as described in Section 5) and increased the learning rate from 1.5×10^{-5} to 1.5×10^{-4}. Our visual observations were again similar to those of EA VAE 3 and EMG VAE, indicating that the the EMG RVQ-VAE network architecture and training dynamics were sufficient for modeling 6 channels of EMG signals.

7 Phase 3: Cross-Modal Mapping & Latent Sequence Generation

Once we had two working RVQ-VAEs, our next challenge was to implement a bridge between these two latent spaces.

Both cross-attention and self-attention are causal in the final MLMLMLM Transformer, to facilitate token-by-token generation at inference time (Section 4.2.3). These features do require adjustments at training time. To simplify debugging, we only implemented these changes to the training script in the last developmental stages of the MLMLMLM Transformer.

Table 5: Comparison of experimental training runs for DGM of EMG signals. The left column shows details about a training run with a vanilla VAE. The right column shows details about a training run with a RVQ-VAE. The 'num conv channels' refers to the number of convolutional channels in both the encoder and the decoder. Wherever a value is —, it is because that metric was not calculated for that training run.

	dataset size	num conv channels	num latent dimensions	KLD β	learning rate	sequence length	codebook size
EMG VAE	augmented	196	64	1×10^{-5}	1.5×10^{-5}	32768	—
EMG RVQ-VAE	augmented	196	64	—	1.5×10^{-4}	65536	2048

Table 6: Comparison of results of experimental training runs for DGM of 6 EMG channels. Wherever a value is —, it is because that metric was not calculated for that training run. All labeled columns are losses averaged over logged training steps within the epoch, except 'epoch' and 'inspection'. The 'inspection' column summarizes the qualitative judgments we made during audio and visual inspection of generated modal outputs.

	epoch	MRSTFT↓	KL loss↓	quantizer↓	adversarial↓	feature-matching↓	disc.↓	train↓	inspection
EMG VAE	160	1.073588	0.003624	—	0.000393	0.019525	1.888853	1.097129	successful
EMG RVQ-VAE	160	**0.711383**	—	0.090453	—	—	—	**0.801835**	successful

We freeze both RVQ-VAEs during training to train only the Transformer. During validation steps, we use the inference pipeline (described in Section 4.2.3), to listen to the final audio outputs of MLMLMLM.

7.0.1 Experiments. First, we trained the Transformer architecture with a self-attention causal mask, but without a cross-attention causal mask. We refer to this experiment as Transformer Experiment 1 (TE1).

Next, we implemented kv caching and a causal mask on the cross-attention mechanism. We made a custom generation function for autoregressive token-by-token generation, where both self-attention and cross-attention are fully causal. We use this function in the validation loop of Transformer Experiment 2 (TE2).

All runs were trained using teacher forcing, though we were able to test sequence continuation with our custom generation scripts during the validation step during training. For both experiments, the vocabulary size is 8192. All runs were trained for 100 epochs. We used Adam optimization and a learning rate of 1×10^{-3} and OneCycleLR annealing with a 15 epoch warm-up.

8 Discussion

The contributions presented herein form part of the first author's ongoing PhD research.[23] A subsequent publication, currently in preparation, will report on artistic practice involving an interactive music system implementation of MLMLMLM.

In terms of future technical directions, we intend to undertake further analysis of the Transformer decoder of MLMLMLM, with particular focus on the strength of the cross-attention conditioning signal. Additionally, there is space to undertake more controlled RVQ-VAE experiments to fully understand the many interdependencies of architectural choices and training dynamics within MLMLMLM. Further exploration could also include systematic cross-modal evaluations and branching to other sensor signal modalities.

Our method contribution and account of our TPR process provide insight about the adaptation of audio DGM architectures to DGM of multi-modal, non-audio sensor signals in NIME. We speculate that our method is most likely to work for scenarios that involve adapting neural audio synthesis architectures to model non-audio time-domain signals, as was our case when designing MLMLMLM.

We attempted to implement a learned start token, so that the sequence generation could be influenced entirely by EMG latents at inference time. However, these attempts were unsuccessful. Our current solution is to encode audio signals (played by a live performer or pre-recorded) at inference time with the audio RVQ-VAE, and use these latents as start tokens for the self-attention mechanism. With this solution, the start token can be conceptualised as a parameter for interactive control through sound in live performance with MLMLMLM.[24]

The most involved DGM task was the modeling of EMG signals because of their lack of interpretability. We had success training the EMG RVQ-VAE with a learning rate of 1.5×10^{-4}, whereas previous vanilla VAE runs with similar channel counts had collapsed with that learning rate. This suggests that vector quantization stabilizes VAE training, which is consistent with literature on RVQ-VAEs [37]. In addition, the absence of KLD in the loss for the EMG RVQ-VAE removed a major competing objective from training, thus simplifying the DGM task.

Since we omitted KLD in our loss for the RVQ-VAEs, the current MLMLMLM architecture does not necessarily support smooth interpolations. Although this functionality is not strictly necessary for the task of muscle-sound machine translation, there is the potential to add these features in the future, towards a more

[23] Research with MLMLMLM is ongoing. Performances and updates are documented on the MLMLMLM project page: https://lucystrauss.com/mlmlmlm

[24] Latency of MLMLMLM is configurable to some extent and is in a trade-off relationship with the number of start-tokens provided at inference time (i.e. MLMLMLM is more likely to generate well given more start-tokens, but the greater the number of start-tokens, the greater the latency). In addition, latency is affected by computing hardware and software signal routing routing of the interactive performance system design in which MLMLMLM is implemented. For these reasons, we set an in-depth discussion of latency aside for a future publication that focuses on live performance with an MLMLMLM-based interactive performance system, whereas this paper focuses on the TPR of our novel DGM method and serves as an initial presentation of the MLMLMLM model, prior to its implementation in an interactive performance system.

versatile model architecture overall. We have not yet tested including KLD during the RVQ-VAE training.

In hindsight, controlled experiments could have been achieved by using RVQ-VAEs for all experiments, rather than vanilla VAEs because findings from one architecture are not necessarily directly transferable to the other. For instance, vector quantization can stabilise VAE training. Furthermore, we spent quite some time tuning KLD β in the loss balance, even though we omitted KLD from the loss function when training the RVQ-VAEs in the current MLMLMLM architecture. Nonetheless, the vanilla VAE experiments were useful for discovering requirements for latent dimensionality and model size. We present our TPR process as it happened in practice, in the hope that process may be informative for fellow NIME researchers.

9 Conclusion

We have contributed a new method for more effectively evaluating DGMs of EMG signals during the NIME design process. This method can hold relevance for any continuous signal with similar temporal structure to audio that has low-interpretability, with the requirement that an interpretable companion modality is present in the dataset.

Our method contribution emerged during the TPR process of MLMLMLM. As such, this paper serves as an example of TPR documentation and strengthens the argument for TPR as a mode of research wherein knowledge emerges through the process of doing technical work.

Complementary to our method contribution, we have contributed MLMLMLM, a novel model architecture for cross-modal Sig2Sig machine translation.[25] The model training supports streaming conditioning at inference time towards integration in live performance settings.

10 Ethical Standards

This research has been approved by the Research Ethics and Integrity Sub-Committee [REISC] Department of Computing Ethics Committee at Goldsmiths, University of London. The dataset was collected from the first author's body and viola sound during a week-long studio residency at the Studios for Electroacoustic Music, Akademie der Kuenste, Berlin. This project takes a first-person research approach and did not involve any participants (other than the first author). This research is supported by the Arts and Humanities Research Council of the United Kingdom and the Consortium for the Humanities and the Arts South-East England.

Acknowledgments

We thank the reviewers for providing immensely helpful reviews.

References

[1] Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. 2023. MusicLM: Generating Music From Text. https://doi.org/10.48550/arXiv.2301.11325 arXiv:2301.11325 [cs].

[2] Ali Mohammad Alqudah and Zahra Moussavi. 2025. Bridging Signal Intelligence and Clinical Insight: A Comprehensive Review of Feature Engineering, Model Interpretability, and Machine Learning in Biomedical Signal Analysis. *Applied Sciences* 15, 22 (Jan. 2025), 12036. https://doi.org/10.3390/app152212036

[3] Daniel Bisig and Kıvanc Tatar. [n. d.]. Raw Music from Free Movements: Early Experiments in Using Machine Learning to Create Raw Audio from Dance Movements. ([n. d.]).

[4] Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. 2023. AudioLM: a Language Modeling Approach to Audio Generation. https://doi.org/10.48550/arXiv.2209.03143 arXiv:2209.03143 [cs].

[5] Antoine Caillon and Philippe Esling. 2021. RAVE: A variational autoencoder for fast and high-quality neural audio synthesis. https://doi.org/10.48550/arXiv.2111.05011 arXiv:2111.05011 [cs, eess].

[6] Franco Caspe, Andrew McPherson, and Mark Sandler. 2025. Waveform Autoencoding at the Edge of Perceivable Latency. (2025).

[7] Chun-Fu (Richard) Chen, Quanfu Fan, and Rameswar Panda. 2021. CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification. 357–366. https://openaccess.thecvf.com/content/ICCV2021/html/Chen_CrossViT_Cross-Attention_Multi-Scale_Vision_Transformer_for_Image_Classification_ICCV_2021_paper.html

[8] João Coimbra, Luís Aly, Henrique Portovedo, Sara Carvalho, and Tiago Bolaños. 2025. EMMA: Enhancing Real-Time Musical Expression through Electromyographic Control. 250–254. https://doi.org/10.5281/zenodo.15698847

[9] Joaquín R Díaz-Durán, Laia Turmo Vidal, and Ana Tajadura-Jiménez. [n. d.]. Joakinator: An Interface for Transforming Body Movement and Perception through Machine Learning and Sonification of Muscle-Tone and Force. ([n. d.]).

[10] Cagri Erdem, Benedikte Wallace, and Alexander Refsum Jensenius. 2022. CAVI: A Coadaptive Audiovisual Instrument–Composition. In *NIME 2022*. PubPub, The University of Auckland, New Zealand. https://doi.org/10.21428/92fbeb44.803c24dd

[11] Çağrı Erdem and Alexander Refsum Jensenius. 2020. RAW: Exploring Control Structures for Muscle-based Interaction in Collective Improvisation. 477–482. https://doi.org/10.5281/zenodo.4813485

[12] Nicholas Evans, Behzad Haki, and Sergi Jordà. [n. d.]. GrooveTransformer: A Generative Drum Sequencer Eurorack Module. ([n. d.]).

[13] Zach Evans, C. J. Carr, Josiah Taylor, Scott H. Hawley, and Jordi Pons. 2024. Fast timing-conditioned latent audio diffusion. In *Forty-first International Conference on Machine Learning*. https://openreview.net/forum?id=jOlO8t1xdx

[14] Zach Evans, Julian D. Parker, C. J. Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. 2024. Stable Audio Open. https://arxiv.org/abs/2407.14358v2

[15] Jamie Ferguson and Stephen A. Brewster. 2017. Evaluation of psychoacoustic sound parameters for sonification. In *Proceedings of the 19th ACM international conference on multimodal interaction (Icmi '17)*. Association for Computing Machinery, Glasgow, UK, 120–127. https://doi.org/10.1145/3136755.3136783 Number of pages: 8 tex.address: New York, NY, USA.

[16] Behzad Haki, Marina Nieto, Teresa Pelinski, and Sergi Jordà Puig. 2022. Real-time drum accompaniment using transformer architecture. (2022). https://doi.org/10.5281/zenodo.7088343

[17] Théo Jourdan and Baptiste Caramiaux. 2023. Machine Learning for Musical Expression: A Systematic Literature Review. (2023). https://hal.science/hal-04075492/

[18] Navroz Kaur Kahlon and Williamjeet Singh. 2023. Machine translation from text to sign language: a systematic review. *Universal Access in the Information Society* 22, 1 (March 2023), 1–35. https://doi.org/10.1007/s10209-021-00823-1

[19] Takatomo Kano, Sakriani Sakti, and Satoshi Nakamura. 2021. Transformer-Based Direct Speech-To-Speech Translation with Transcoder. In *2021 IEEE Spoken Language Technology Workshop (SLT)*. 958–965. https://doi.org/10.1109/SLT48900.2021.9383496

[20] SangYeon Kim, Hyunwoo Lee, Jonghee Han, and Joon-Ho Kim. 2021. Sig2Sig: Signal Translation Networks to Take the Remains of the Past. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 3620–3624. https://doi.org/10.1109/ICASSP39728.2021.9415084 ISSN: 2379-190X.

[21] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. 2022. Autoregressive Image Generation Using Residual Quantization. 11523–11532. https://openaccess.thecvf.com/content/CVPR2022/html/Lee_Autoregressive_Image_Generation_Using_Residual_Quantization_CVPR_2022_paper.html

[22] Davide Lionetti, Luca Turchet, Massimiliano Zanoni, and Paolo Belluco. 2024. Muscle-Guided Guitar Pedalboard: Exploring Interaction Strategies Through Surface Electromyography and Deep Learning. 241–251. https://doi.org/10.5281/zenodo.13904842

[23] Jeffrey A. T. Lupker. 2021. Score-Transformer: A Deep Learning Aid for Music Composition. https://doi.org/10.21428/92fbeb44.21d4fd1f

[24] Pasquale Mainolfi. 2025. Simulated EEG-Driven Audio Information Mapping Using Inner Hair-Cell Model and Spiking Neural Network. (2025).

[25] Thomas Nuttall, Behzad Haki, and Sergi Jorda. 2021. Transformer Neural Networks for Automated Rhythm Generation. *International Conference on New Interfaces for Musical Expression* (April 2021). https://doi.org/10.21428/92fbeb44.fe9a0d82

[26] Evan O'Donnell and Patrick Gunawan Hartono. 2025. A Practice-Based Methodology for Capturing Embodied Gesture-Rhythm Relations in Small Datasets. (Sept. 2025). https://research-repository.rmit.edu.au/articles/conference_contribution/A_Practice-Based_Methodology_for_Capturing_Embodied_Gesture-Rhythm_Relations_in_Small_Datasets/30443165/1

[25] We share the open-source MLMLMLM codebase on GitHub https://github.com/lucystrauss/MLMLMLM

[27] Teresa Pelinski, Rodrigo Diaz, Adán L. Benito Temprano, and McPherson McPherson. 2023. Pipeline for recording datasets and running neural networks on the Bela embedded hardware platform. In *NIME 2023*. Mexico City, Mexico.

[28] Teresa Pelinski, Andrew McPherson, and Rebecca Fiebrink. 2025. Ways of knowing, ways of writing: technical practice research in new musical instrument design. *Journal of New Music Research* (Jan. 2025), 1–14. https://doi.org/10.1080/09298215.2024.2442348

[29] Teresa Pelinski, Giulio Moro, and Andrew McPherson. 2025. pybela: a Python library to interface scientific and physical computing. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. 63–72. https://nime.org/proceedings/2025/nime2025_9.pdf

[30] Thierry Poibeau. 2017. *Machine Translation*. MIT Press. Google-Books-ID: LYc3DwAAQBAJ.

[31] Courtney Reed and Andrew McPherson. 2020. Surface Electromyography for Direct Vocal Control. 458–463. https://doi.org/10.5281/zenodo.4813475

[32] P.J. Sparto, M. Parnianpour, E.A. Barria, and J.M. Jagadeesh. 2000. Wavelet and short-time Fourier transform analysis of electromyography for detection of back muscle fatigue. *IEEE Transactions on Rehabilitation Engineering* 8, 3 (Sept. 2000), 433–436. https://doi.org/10.1109/86.867887

[33] Lucy Strauss and Matthew Yee-King. 2023. Extensible Embodied Knowledge: Bridging Performance Practice and Intelligent Performance System Design. In *AIMC 2023*. https://research.gold.ac.uk/id/eprint/34056/

[34] Lucy Strauss and Matthew Yee-King. 2024. Towards a Machine Somaesthete: Latent Modeling of EMG Signals in Viola Playing. In *9th International Conference on Movement and Computing*. Utrecht, Netherlands, 11 pages. https://doi.org/10.1145/3658852.3659074

[35] Jung-Hoon Sul, Lasitha Piyathilaka, Diluka Moratuwage, Sanura Dunu Arachchige, Amal Jayawardena, Gayan Kahandawa, and D. M. G. Preethichandra. 2025. Electromyography Signal Acquisition, Filtering, and Data Analysis for Exoskeleton Development. *Sensors (Basel, Switzerland)* 25, 13 (June 2025), 4004. https://doi.org/10.3390/s25134004

[36] Kıvanç Tatar, Daniel Bisig, and Philippe Pasquier. 2021. Latent Timbre Synthesis. *Neural Computing and Applications* 33, 1 (Jan. 2021), 67–84. https://doi.org/10.1007/s00521-020-05424-2

[37] Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. 2017. Neural Discrete Representation Learning. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/hash/7a98af17e63a0ac09ce2e96d03992fbc-Abstract.html

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[39] Tony Veale, Alan Conway, and BrÓna Collins. 1998. The Challenges of Cross-Modal Translation: English-to-Sign-Language Translation in the Zardoz System. *Machine Translation* 13, 1 (March 1998), 81–106. https://doi.org/10.1023/A:1008014420317

[40] Felipe Verdugo, Amedeo Ceglia, Christian Frisson, Alexandre Burton, Mickael Begon, Sylvie Gibet, and Marcelo M. Wanderley. 2022. Feeling the Effort of Classical Musicians - A Pipeline from Electromyography to Smartphone Vibration for Live Music Performance. https://doi.org/10.21428/92fbeb44.3ce22588

[41] Prateek Verma and Chris Chafe. 2021. A Generative Model for Raw Audio Using Transformer Architectures. In *2021 24th International Conference on Digital Audio Effects (DAFx)*. IEEE, Vienna, Austria, 230–237. https://doi.org/10.23919/DAFx51585.2021.9768298

[42] Haifeng Wang, Hua Wu, Zhongjun He, Liang Huang, and Kenneth Ward Church. 2022. Progress in Machine Translation. *Engineering* 18 (Nov. 2022), 143–153. https://doi.org/10.1016/j.eng.2021.03.023

[43] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2022. SoundStream: An End-to-End Neural Audio Codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 30 (2022), 495–507. https://doi.org/10.1109/TASLP.2021.3129994

[44] Liang Zhao, Xiachong Feng, Xiaocheng Feng, Weihong Zhong, Dongliang Xu, Qing Yang, Hongtao Liu, Bing Qin, and Ting Liu. 2024. Length Extrapolation of Transformers: A Survey from the Perspective of Positional Encoding. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 9959–9977. https://doi.org/10.18653/v1/2024.findings-emnlp.582

[45] Liu Ziyin, Tilman Hartwig, and Masahito Ueda. 2020. Neural Networks Fail to Learn Periodic Functions and How to Fix It. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 1583–1594. https://proceedings.neurips.cc/paper/2020/hash/1160453108d3e537255e9f7b931f4e90-Abstract.html

A Spectrogram Reconstructions from an Unsuccessful Training Experiment

The following spectrograms are from the 'EA VAE 1' experimental training run detailed in Section 6.1. The model failed to generate a variety of model outputs. Notice that all generated outputs are visually indistinguishable, suggesting convergence to a single average-like solution. For comparison, we share spectrograms from a successful training run in Appendix B & C.[26]

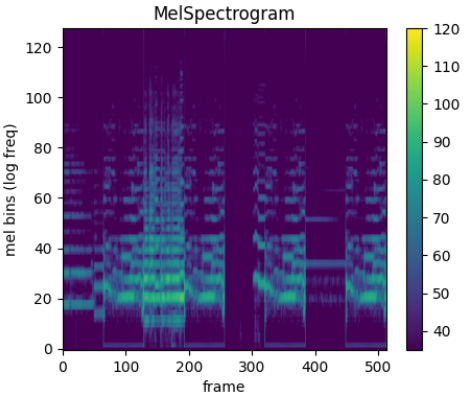


Figure 7: Audio Channel. This spectrogram depicts four pairs of original and reconstructed signals. Original and reconstructed samples are interleaved.

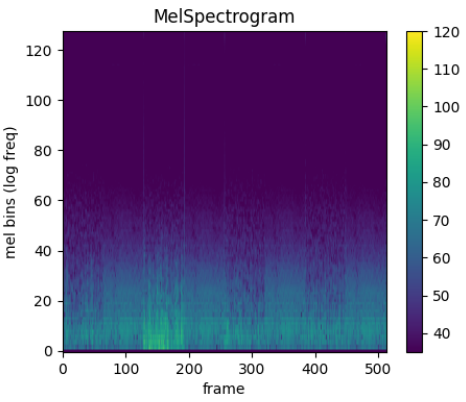


Figure 8: EMG Channel 1. This spectrogram depicts four pairs of original and reconstructed signals. Original and reconstructed samples are interleaved.

[26] Associated listening examples are available online: https://lucystrauss.github.io/NIME_2026_examples/

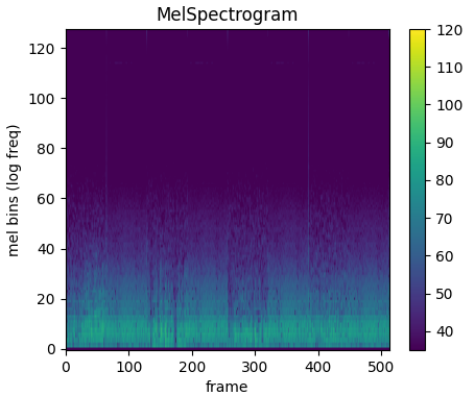


Figure 9: EMG Channel 2. This spectrogram depicts four pairs of original and reconstructed signals. Original and reconstructed samples are interleaved.

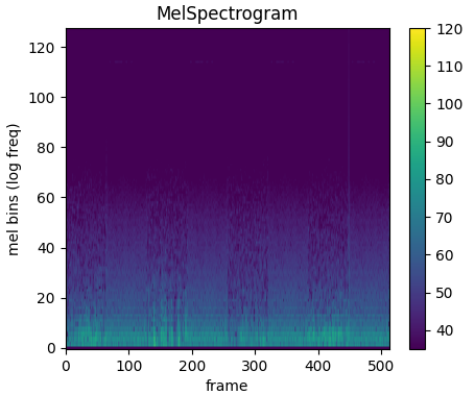


Figure 10: EMG Channel 3. This spectrogram depicts four pairs of original and reconstructed signals. Original and reconstructed samples are interleaved.

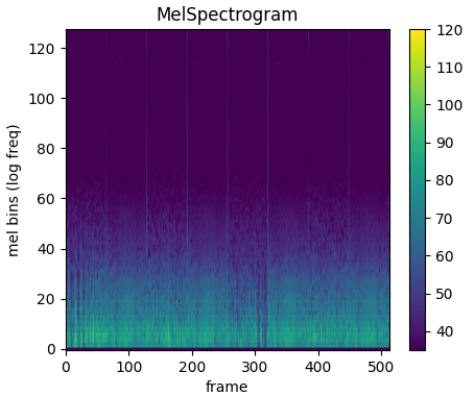


Figure 11: EMG Channel 4. This spectrogram depicts four pairs of original and reconstructed signals. Original and reconstructed samples are interleaved.

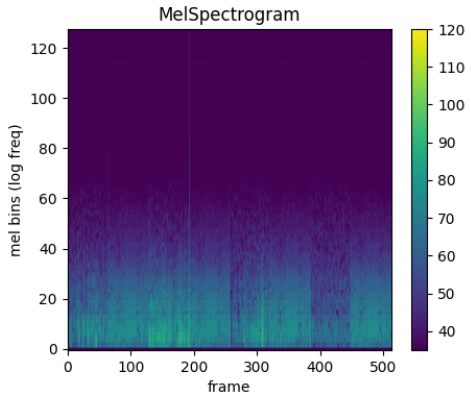


Figure 12: EMG Channel 5. This spectrogram depicts four pairs of original and reconstructed signals. Original and reconstructed samples are interleaved.

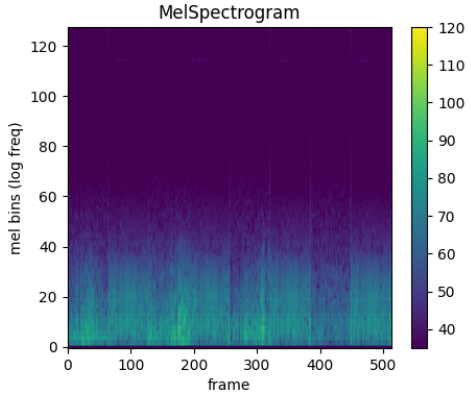


Figure 13: EMG Channel 6. This spectrogram depicts four pairs of original and reconstructed signals. Original and reconstructed samples are interleaved.

B Spectrogram Reconstructions from a Successful Training Experiment

The spectrograms in this section are from the 'EA VAE 3' experimental training run (Section 6.1) at 113 epochs. The model was able to generate a variety of model outputs.

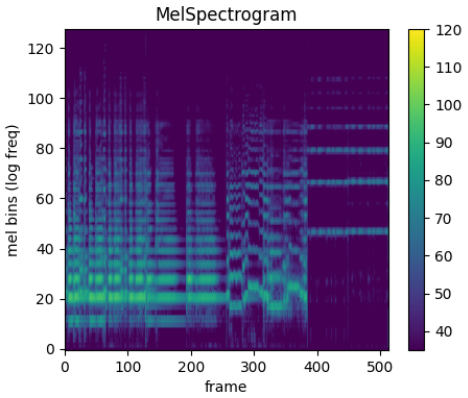


Figure 14: Audio Channel. This spectrogram depicts four pairs of original and reconstructed signals. Original and reconstructed samples are interleaved.

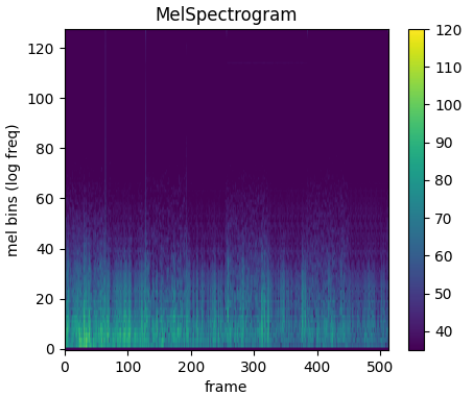


Figure 15: EMG Channel 1. This spectrogram depicts four pairs of original and reconstructed signals. Original and reconstructed samples are interleaved.

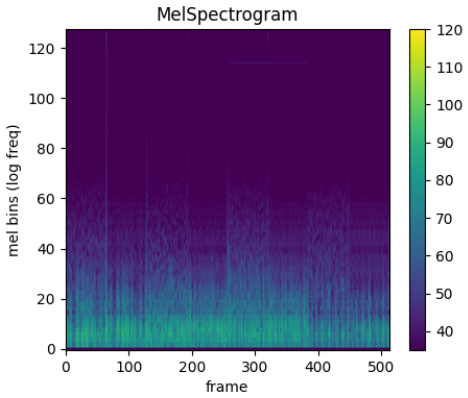


Figure 16: EMG Channel 2. This spectrogram depicts four pairs of original and reconstructed signals. Original and reconstructed samples are interleaved.

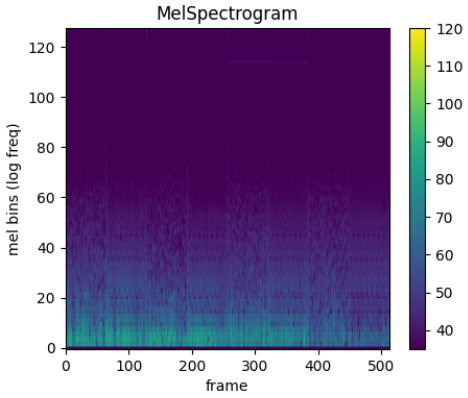


Figure 17: EMG Channel 3. This spectrogram depicts four pairs of original and reconstructed signals. Original and reconstructed samples are interleaved.

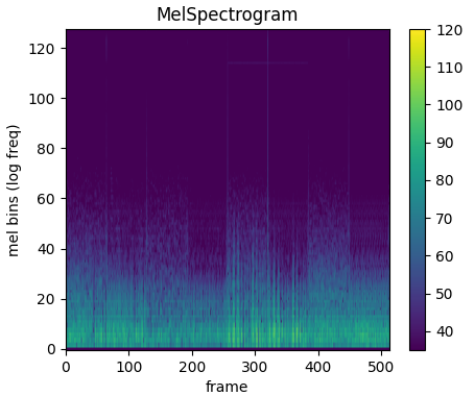


Figure 18: EMG Channel 4. This spectrogram depicts four pairs of original and reconstructed signals. Original and reconstructed samples are interleaved.

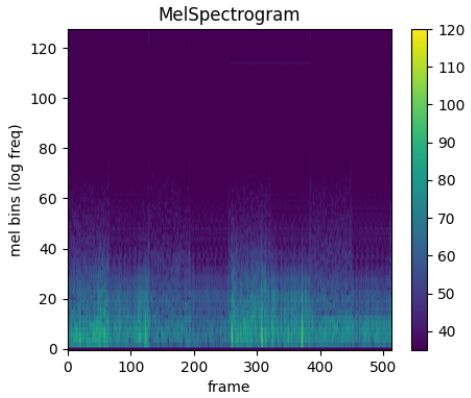


Figure 19: EMG Channel 5. This spectrogram depicts four pairs of original and reconstructed signals. Original and reconstructed samples are interleaved.

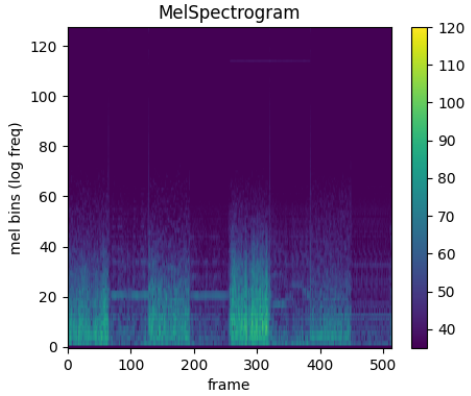


Figure 20: EMG Channel 6. This spectrogram depicts four pairs of original and reconstructed signals. Original and reconstructed samples are interleaved.

C Improvements to Reconstructions after More Training

The paper body reports loss values at 113 epochs for comparison between experiments. However, we noticed during training that this run seemed promising and continued training 'EA VAE 3' for 175 epochs total. Reconstruction accuracy seems to have improved for channel 6 in particular, compared to Appendix B.

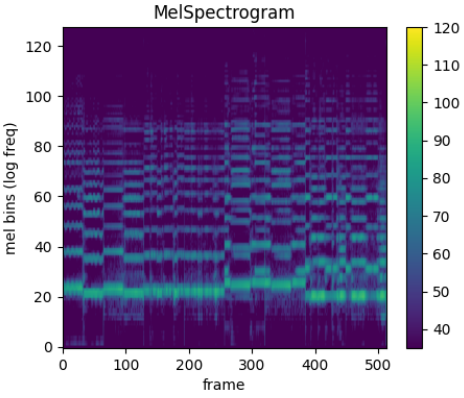


Figure 21: Audio Channel. This spectrogram depicts four pairs of original and reconstructed signals. Original and reconstructed samples are interleaved.

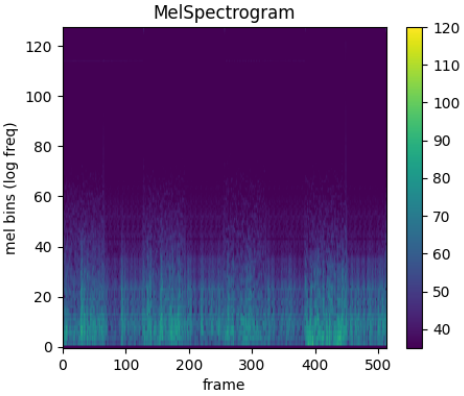


Figure 22: EMG Channel 1. This spectrogram depicts four pairs of original and reconstructed signals. Original and reconstructed samples are interleaved.

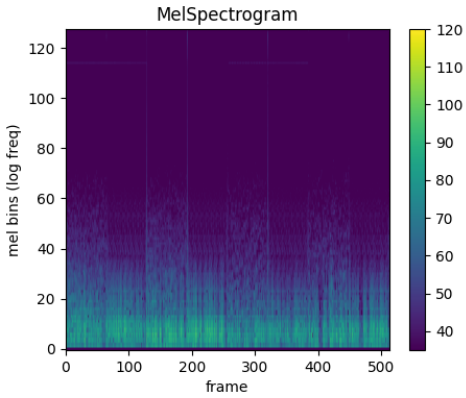


Figure 23: EMG Channel 2. This spectrogram depicts four pairs of original and reconstructed signals. Original and reconstructed samples are interleaved.

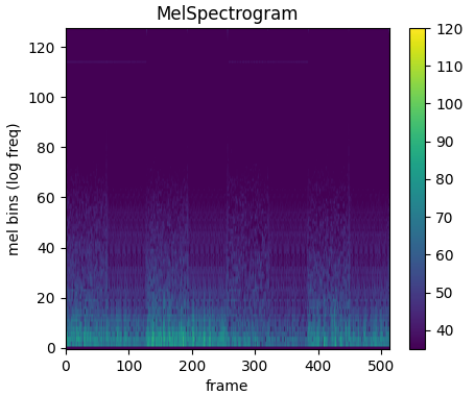


Figure 24: EMG Channel 3. This spectrogram depicts four pairs of original and reconstructed signals. Original and reconstructed samples are interleaved.

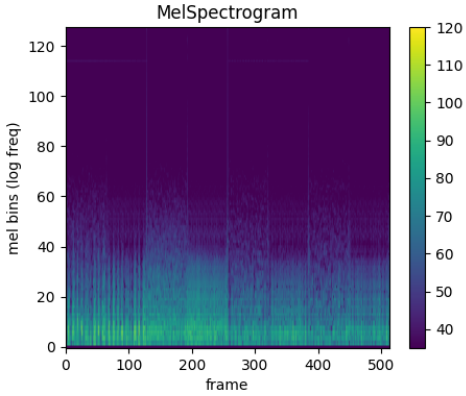


Figure 25: EMG Channel 4. This spectrogram depicts four pairs of original and reconstructed signals. Original and reconstructed samples are interleaved.

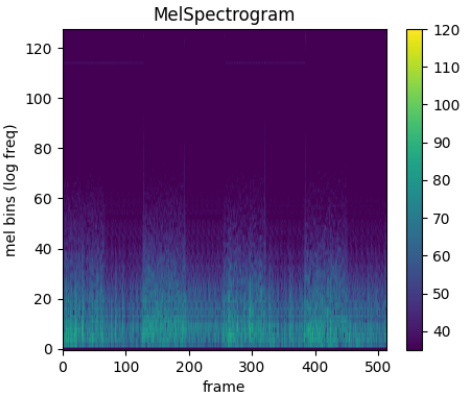


Figure 26: EMG Channel 5. This spectrogram depicts four pairs of original and reconstructed signals. Original and reconstructed samples are interleaved.

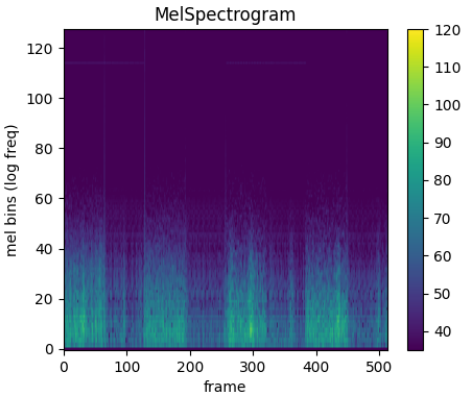


Figure 27: EMG Channel 6. This spectrogram depicts four pairs of original and reconstructed signals. Original and reconstructed samples are interleaved.