

GENERATING SAMPLE-BASED MUSICAL INSTRUMENTS USING NEURAL AUDIO CODEC LANGUAGE MODELS

Shahan Nercessian* Johannes Imort* Ninon Devis Frederik Blang
Native Instruments

firstname.lastname@native-instruments.com

*Equal contribution

ABSTRACT

In this paper, we propose and investigate the use of neural audio codec language models for the automatic generation of sample-based musical instruments based on text or reference audio prompts. Our approach extends a generative audio framework to condition on pitch across an 88-key spectrum, velocity, and a combined text/audio embedding. We identify maintaining timbral consistency within the generated instruments as a major challenge. To tackle this issue, we introduce three distinct conditioning schemes. We analyze our methods through objective metrics and human listening tests, demonstrating that our approach can produce compelling musical instruments. Specifically, we introduce a new objective metric to evaluate the timbral consistency of the generated instruments and adapt the average Contrastive Language-Audio Pretraining (CLAP) score for the text-to-instrument case, noting that its naive application is unsuitable for assessing this task. Our findings reveal a complex interplay between timbral consistency, the quality of generated samples, and their correspondence to the input prompt.

1. INTRODUCTION

The exploration of sound synthesis and the development of interfaces to manipulate timbre are fundamental topics in audio research [1]. With the evolution of sound synthesis in the digital realm, musicians have unprecedented means to manifest their artistic visions. Meanwhile, generative models for images and text have shown disruptive abilities in creating novel samples from learned distributions [2]. It becomes only natural to consider implications of such technologies when applied to a music production context.

Several generative models for neural audio synthesis have been put forth, including NSynth [3], which uses a WaveNet [4] autoencoder to create samples of pitched instruments, and GANSynth [5], which models signal phase through an instantaneous frequency representation. Furthermore, Differentiable Digital Signal Processing (DDSP)

[6] and its related works [7] introduce autoencoders with differentiable synthesizers for improved controllability, while a novel approach via a real-time variational autoencoder is presented in [8]. Additionally, GANstrument [1] leverages a feature descriptor obtained through adversarial domain confusion, highlighting the diverse methodologies employed to advance the field of audio synthesis. These models lack an interface for controlling audio generation via text input. Accordingly, we have witnessed a surge in text-to-audio systems generating convincing audio examples from text prompts [9]. One family of approaches rely on neural audio codecs [10, 11] representing audio as a set of discrete codes whose sequence can be learned using transformer-based language models. While initial approaches targeted speech [12, 13] and ambient sounds [14], follow-on works adapt methods for text-to-music generating full musical passages from text [15, 16].

Though compelling, seminal text-to-music works target generation of entire musical arrangements or otherwise lack fine-grained control over their outputs, and might not integrate well into musicians' workflows. Consequently, efforts have been made to adapt these models to sit closer in the creative process. These include StemGen [17], predicting instrument track layers from a given musical context, and VampNet [18], generating musical variations via generative filling. We align with this philosophy, intending to enable new sounds to inspire musical creativity.

In this paper, we introduce the application of neural audio codec language models for the automated creation of sample-based musical instruments using both text and audio prompts as input, building upon our preliminary work in progress in [19]. We model a musical instrument as a set of waveforms sampling the instrument's time-domain response across the dimensions of pitch (the fundamental frequency of a note) and velocity (the intensity with which a note is played). Under this paradigm, we move beyond the constraints of any one parametric synthesizer, avoiding expressivity limitations tied to its implementation. As in [1], we note that injecting inductive bias into the generative process via DDSP is interesting but complementary to our work, as such methods constrain the manifold that outputs can live on [20]. Unlike text-to-music systems, which typically generate a single audio example for a given text prompt during inference, prompt-to-instrument systems must generate an ensemble of audio samples from a fixed prompt, which must be pitch-accurate and timbrally



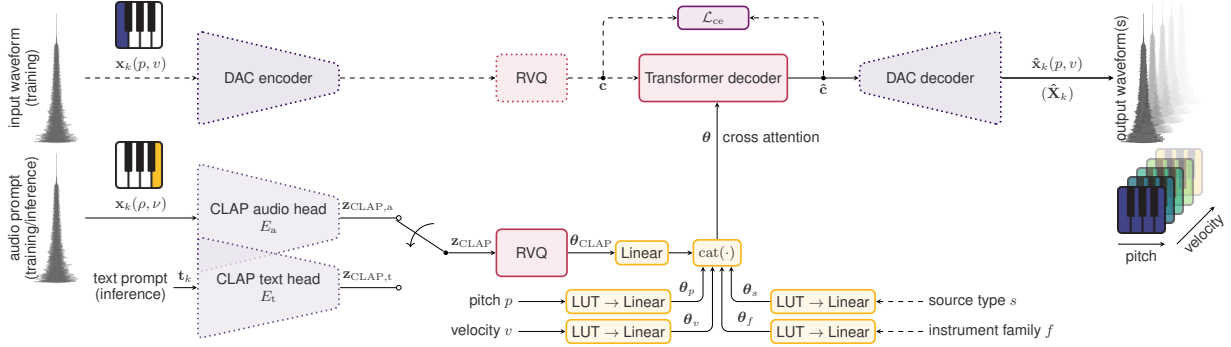


Figure 1. Overview of our proposed system. Dotted lines represent frozen pretrained modules. Dashed lines denote steps exclusive to training. CLAP’s audio or text head can be used at inference, disregarding source type and instrument family. Training operates on individual samples \mathbf{x} , while inference creates a set of samples $\hat{\mathbf{X}}$ from a consistent CLAP prompt and varied pitch/velocity cues to create a full instrument. Different piano keys/colors denote different pitches/velocities.

consistent with one another to allow for the assembly of a playable instrument. Our contributions are as follows:

- We introduce the text-to-instrument (T2I) task, in which waveforms comprising a sample-based musical instrument are generated from a user text prompt.
- We propose neural audio codec language models as solutions for both text- and audio-prompted sample-based instrument generation, expanding on a state-of-the-art generative audio model that is conditioned on a Contrastive Language-Audio Pretraining (CLAP) embedding [21], as well as pitch across the 88-key range of a standard full-length piano keyboard, velocity, instrument family and source type.
- We develop an objective metric to assess the timbral consistency (TC) of sample-based instruments.
- We propose an adaptation to the average CLAP score to be suitable for objectively assessing T2I.
- We propose and analyze three CLAP conditioning schemes through qualitative and quantitative means.
- We demonstrate the compatibility of our approach with both autoregressive (AR) and non-AR audio transformers like MAGNeT [22].

The remainder of this paper is organized as follows: Section 2 describes our proposed method, Section 3 outlines quantitative metrics for assessing performance, including the ones that we have developed, Section 4 reports our experimental results, and Section 5 draws conclusions.

2. PROPOSED METHOD

Figure 1 illustrates our proposed method, which is based on MusicGen [16] as a foundation, consisting of a neural audio codec and a language model to predict acoustic tokens from conditioning signals. We replace EnCodec [23] used in MusicGen with the Descript Audio Codec (DAC) [11], addressing codebook collapse in previous models while achieving higher audio fidelity. We also introduce a set of new conditioning signals including pitch and velocity, alongside a CLAP embedding [21]. Our conditioning signals reflect global cues θ for steering generation, which are fused with the language model via cross-attention. Using CLAP allows instrument samples to be inferred from

either audio or text prompts, and we denote their tasks as sample-to-instrument (S2I) and T2I, respectively. The aim of S2I may be considered one of pitch/velocity shifting, whereby the model transforms an audio prompt in ways transcending conventional signal processing. In T2I, text acts as a semantic interface to generate instruments whose timbres may otherwise not exist. To ensure the reproducibility of our findings, we use pretrained sub-networks without modification, training our core language models from random initialization on the standard research dataset NSynth [3]. We acknowledge that fine-tuning sub-modules within a generative model can improve a composite system, but consider this to be outside the scope of this work.

2.1 Compressed audio representation

We use the DAC encoder to create an intermediate representation of a monophonic waveform \mathbf{x} , resulting in the discrete codes \mathbf{c} , while the DAC decoder synthesizes an audio waveform $\hat{\mathbf{x}}$ from a predicted code sequence $\hat{\mathbf{c}}$. The DAC is trained on a broad spectrum of audio types, so we deem it suitable for generating tonal one-shot instrumental sounds. We model our task at a sample rate of 44.1 kHz, as this would be a minimum requirement for real-world music production use cases. We employ the corresponding pretrained model with fixed weights during training.

2.2 Language model

To model the discrete audio tokens of single-shot samples, we consider a smaller, 60M parameter variant of the transformer decoder in [16], in order to prevent overfitting, speed up inference, and conceptually demonstrate our approach. The model consists of 12 layers with 16 attention heads per layer and a transformer dimension $d = 512$. We consider scaling our models to larger sizes to be out of scope for this work. As in MusicGen [16], we predict audio from tokens of the 4 most significant [11] codebooks at each frame (of the 9 supported by DAC), selecting tokens from codebooks of size 1024. At inference time, we consider next-token prediction using AR sampling with delayed pattern interleaving [16], as well as the iterative decoding scheme proposed in [22] reporting a $7\times$ inference

speed-up. For MAGNeT-style inference, we use 20 decoding steps for the first codebook, and 10 for the remaining codebooks, respectively (compared to 345 steps for the AR scheme). As is customary, we can leverage classifier-free guidance at inference time in both cases [16, 17]. We expect AR priors to provide higher fidelity, considering the importance of onsets to perception [24] for the single-shot samples that we generate: earlier audio token predictions are likely to be perceptually more relevant than later ones.

2.3 Categorical conditioning

We use a categorical conditioning scheme for pitch p , velocity v , broad instrument family f , and source type s , that consists of a lookup table (LUT) and a fully connected layer that maps the dimension of the categorical feature space to the dimension d of the language model. For pitch, we model the $d_p = 88$ range of notes spanned by a full-length keyboard, corresponding to Musical Instrument Digital Interface (MIDI) note numbers 21-108, and note this to be a significant expansion relative to the chroma feature used in [16]. We consider $d_v = 5$ velocity layers, according to MIDI velocities 25, 50, 75, 100, and 127 within our training dataset. The instrument family (i.e. bass, brass, etc.) and source type (i.e., acoustic, electronic, etc.) attributes in our dataset serve as metadata-driven timbral cues that could optionally guide training [25], but we do not expect them to be specified at inference. We choose to include them for models trained in this work, subjecting them to dropout with 30% probability, noting that dropout can generalize their complete inclusion or exclusion.

2.4 Joint text and audio conditioning

We use the CLAP model [21], employing encoders to generate a common fixed-dimensional representation for audio/text pairs of size $d_z = 512$. This model was pretrained on musical signals, utilizing a contrastive loss to align respective audio and text embeddings, ultimately enabling the use of either modality as input to our system. The audio encoder E_a uses HTS-AT [26], while the text encoder E_t is based on RoBERTa [27]. Given an audio dataset of instrumental samples, this strategy allows for leveraging only the audio head during language model training, without requiring rich text captions in the dataset. We quantize resulting CLAP embeddings through Residual Vector Quantization (RVQ) with learned codes [16], yielding θ_{CLAP} .

A distinction between generating music and creating sample-based instruments from prompts is that the inference scenario for instrument generation utilizes a single fixed representation as input for generating a cohesive set of waveforms comprising an instrument. Consequently, we present three CLAP conditioning schemes specifically to train language models for sample-based instrument creation. These techniques amount to assigning pairs of $\mathbf{z}_{\text{CLAP},a}$ and codes \mathbf{c} as input and target training examples in various ways, where $\mathbf{z}_{\text{CLAP},a}$ is the output of the CLAP audio encoder E_a . Hence, the target codes and CLAP embedding within a training example need not be derived from the same waveform, so long as they come

from the same instrument. Excluding θ_f and θ_s for clarity, the forward pass observed during the training of a language model Θ is

$$\hat{\mathbf{c}} = \Theta(\mathbf{z}_{\text{CLAP},a}, \theta_p, \theta_v), \quad (1)$$

where $\mathbf{z}_{\text{CLAP},a} = E_a(\mathbf{x}_k(\rho, \nu))$. Here, k , ρ , and ν denote the timbre (i.e. instrument), pitch, and velocity exhibited in an underlying audio example, respectively, which we assume to be readily selectable from our training set. This $\mathbf{x}_k(\rho, \nu)$ is the input to E_a , and need not be identical to $\mathbf{x}_k(p, v)$ which is used to derive the target codes \mathbf{c} .

2.4.1 Baseline CLAP

By design, the CLAP audio encoder E_a will inevitably yield distinct numerical representations for instrumental samples of the same instrument but varying in pitch or velocity. During training, the following equation applies:

$$\mathbf{z}_{\text{CLAP},a} = E_a(\mathbf{x}_k(p, v)), \quad (2)$$

While this suffices for creating a music track from a singular representation, the scenario diverges significantly for sample-based instrument creation. Specifically, pitch and velocity are represented through both the CLAP representation as well as their respective categorical conditioners, which can reduce the overall effectiveness of the latter. We consider this adaptation of existing prompt-to-audio methodologies to serve as a baseline in this work, noting its application to this task is still novel.

2.4.2 Random CLAP

In order to disentangle the aforementioned pitch/velocity effect, we consider a randomization technique defined by

$$\mathbf{z}_{\text{CLAP},a} = E_a(\mathbf{x}_k(\tilde{\rho}, \tilde{\nu})), \quad (3)$$

with $\tilde{\rho} \sim \mathcal{U}\{21, \dots, 108\}$, and $\tilde{\nu} \sim \mathcal{U}\{25, 50, 75, 100, 127\}$. Random selection with replacement is performed throughout training. This method resembles the nearest neighbor data augmentation in [1], where we consider samples to be neighbors if they originate from the same instrument.

2.4.3 Fixed CLAP

Lastly, we consider a conditioning scheme where we use a fixed, predefined CLAP embedding for each instrument as

$$\mathbf{z}_{\text{CLAP},a} = E_a(\mathbf{x}_k(\rho_{0,f}, \nu_0)), \quad (4)$$

where $\rho_{0,f}$ is defined for each instrument family f (see Table 1) such that fixed representations are sampled within the natural range of each instrument (i.e. we make

Instrument families	Note name
Bass	C2
Brass, String, Synth lead	C3
Guitar, Keyboard, Organ, Reed, Vocal	C4
Flute, Mallet	C5

Table 1. Pitch values used for fixed CLAP conditioning.

lower-pitched selections for bass sounds). The categorical velocity ν_0 is fixed across the training set at velocity 100, conveying an instrument’s timbre played with a medium/strong intensity. If a sample matching a $\rho_{0,f}$ and ν_0 query is not available within an instrument, we opt for its nearest available pitch, followed by its nearest velocity.

Other fixed CLAP conditioning forms could also have been devised, e.g. using average per-instrument CLAP embeddings. We opt for our described approach as it ensures that each CLAP embedding used in model training originates from exactly one audio example. We assert that this fixed variant most closely aligns training to the scenario at inference. In fact, we posit that both the baseline and random CLAP approaches are data augmentation alternatives relative to this method, that increase the number of conditioning signal/target code pairs observed during training, while potentially introducing domain mismatches.

3. OBJECTIVE EVALUATION CRITERIA

We assess models across several objective criteria for S2I and T2I. Alongside the widely used Fréchet audio distance (FAD) [28] score, we introduce a novel metric to evaluate the TC of generated sample-based instruments. We also propose an adaptation of the average CLAP score to fairly evaluate text correspondence for T2I. Unless otherwise specified, we base instrument generation-specific metrics on the assumption that they are represented by $N_k = d_p d_v = 440$ audio samples. In practice, care is taken to properly aggregate/mask instrument statistics based on which samples are present.

3.1 FAD score

The FAD score allows a common framework for evaluating generative audio models using almost any audio feature descriptor [28]. We utilize a FAD metric formulated using VGGish, as in related works [15, 17]. We also report FAD scores using CLAP (audio) embeddings, since they form a pivotal component to our system, allow analysis for higher-sample rate audio (48 kHz), and have been shown to have increased correlation to perception relative to VGGish [29]. The FAD score is generically defined as

$$\text{FAD}(\mathbf{Z}_1, \mathbf{Z}_2) = \|\mu_1 - \mu_2\|_2^2 + \text{Tr}\left(\mathbf{A}_1 + \mathbf{A}_2 + (\mathbf{A}_1 \mathbf{A}_2)^{\frac{1}{2}}\right), \quad (5)$$

where $\mathbf{Z}_i \in \mathbb{R}^{d_z \times TN}$ is a collection of T d_z -dimensional embeddings extracted by a given audio descriptor, across N samples from a population $i \in [1, 2]$. Considering the 4-second long audio segments generated in this work and the strides of various models, $T = 4$ and 1 when using VGGish and CLAP, respectively. We reserve subscripts 1 and 2 to denote ground truth/test populations, respectively. Accordingly, each \mathbf{Z}_i has mean $\mu_i \in \mathbb{R}^{d_z}$ and covariance $\mathbf{A}_i \propto \mathbf{Z}_i \mathbf{Z}_i^\top \in \mathbb{R}^{d_z \times d_z}$. The first and second terms in Equation 5 quantify mean correspondence and similarities in the spread between distributions, respectively. The FAD score possesses a property allowing unpaired populations

to be compared, which we use as a criterion to assess "in-the-wild" T2I in lieu of ground truth audio.

3.2 TC score

Our system should generate timbrally consistent samples in order for them to triggered harmoniously as a sample-based instrument, and we aim to characterize this quantitatively. An apt definition for TC may seem ill-posed, since we want instrument samples to be fundamentally consistent with one another, but also expect them to exhibit some timbral variations as functions of pitch/velocity. This is particularly sought-after in high-quality virtual instruments, motivating the modeling approach in [6]. To contend with these potentially conflicting aspirations, we learn statistics from existing sample-based instruments serving as prototypes for realistic TC, and build metrics around them. We use CLAP embeddings as a basis to create an elegant embodiment in this work. To do so, we forego the mean subtraction step standard to covariance matrix computations, noting that samples are practically close to zero-mean in this respect. Hereafter, we use the terms covariance, affinity, and cosine similarity interchangeably.

We define per-instrument covariance matrices as

$$\mathbf{A}_{ij,k} = \frac{1}{N_k} \mathbf{Z}_{i,k}^\top \mathbf{Z}_{j,k}, \quad (6)$$

where $\mathbf{A}_{ij,k} \in \mathbb{R}^{N_k \times N_k}$ is the affinity between embeddings $\mathbf{Z}_{i,k}$ and $\mathbf{Z}_{j,k} \in \mathbb{R}^{d_z \times N_k}$ representing the subset of CLAP embeddings of the k th instrument within each population. Here, we compute statistics emphasizing variations across samples instead of feature dimensions. Referring to Equation 5, the L_2 -normalized quality CLAP embeddings will ensure us that $\text{Tr}(\mathbf{A}_{ii,k}) = 1 \forall i \in [1, 2]$ and $k \in [1, \dots, K]$. Accordingly, we can define

$$\text{TC}_{\text{CLAP}}(\mathbf{Z}_1, \mathbf{Z}_2) = \frac{1}{K} \sum_k \text{Tr}\left((\mathbf{A}_{11,k} \mathbf{A}_{22,k})^{\frac{1}{2}}\right), \quad (7)$$

which is bounded in $[0, 1]$ and aggregates the similarity in covariations across instruments within each population. Instead of using $\mathbf{A}_{11,k}$ for making comparisons between populations on a per-instrument basis, we consider $\mathbf{A}_{11,*} = \frac{1}{K} \sum_k \mathbf{A}_{11,k}$, averaging per-instrument affinity matrices across a ground truth evaluation set. This provides richer statistics for improved stability, and a unified method to assess TC for S2I and T2I. The TC score is then

$$\text{TC}_{\text{CLAP}^*}(\mathbf{Z}_1, \mathbf{Z}_2) = \frac{1}{K} \sum_k \text{Tr}\left((\mathbf{A}_{11,*} \mathbf{A}_{22,k})^{\frac{1}{2}}\right). \quad (8)$$

We compute $\mathbf{A}_{11,*}$ using all of the samples from the NSynth validation and test sets that are within our desired 88-key pitch range, reflecting a total of 53 instruments. The resulting covariance matrix is illustrated in Figure 2c, in which samples are ordered primarily by pitch and secondarily by velocity. Note how $\mathbf{A}_{11,*}$ deviates from "ideal TC," whereby all embeddings would be correlated with unity similarity (see Figure 2a). Moreover, a 5×5 texture

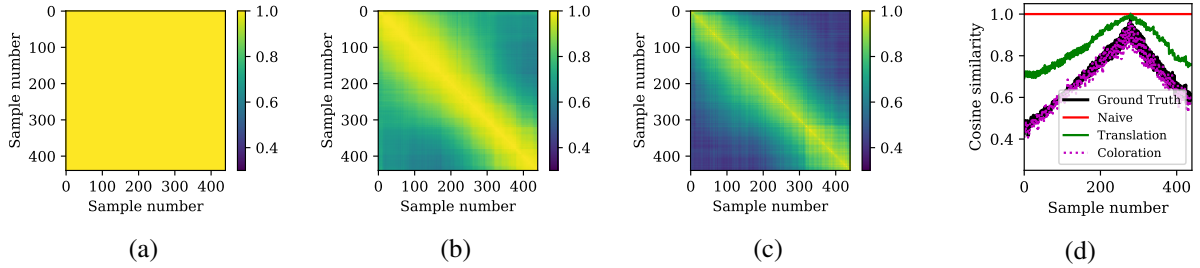


Figure 2. Covariance matrices for the text prompt $\mathbf{t}_k = \text{aggressive synth lead}$, computed using (a) naive replication, (b) translation, (c) coloration (matching the ground truth covariance $\mathbf{A}_{11,*}$ learned over the 53 instruments reflected in the NSynth validation/test sets), (d) cosine similarities relative to estimated $\hat{\rho}_k/\hat{\nu}_k$, corresponding to note E5/velocity 100.

emerges in $\mathbf{A}_{11,*}$, indicative of variations in cosine similarity amongst samples of the same pitch but differing velocities. Lastly, one may question the suitability of CLAP as a feature descriptor within this context, given its variability concerning pitch/velocity discussed in Section 2.4. Its improved correlation to perception aside [29], we assert that learning statistics over data effectively embeds potential measurement deficiencies that effectively neutralizes when we compare new population statistics against it.

3.3 Average CLAP score

3.3.1 Sample-to-instrument (S2I)

Given $N = \sum_k N_k$ and a cross-population covariance $\mathbf{A}_{ij} = \frac{1}{N} \mathbf{Z}_i^\top \mathbf{Z}_j \in \mathbb{R}^{N \times N}$, the average CLAP score computed on a per-sample basis can be expressed concisely as

$$s_{\text{CLAP}}(\mathbf{Z}_1, \mathbf{Z}_2) = \text{Tr}(\mathbf{A}_{12}) = \frac{1}{N} \sum_k N_k \text{Tr}(\mathbf{A}_{12,k}). \quad (9)$$

It can also be computed on a per-instrument basis by

$$s_{\text{CLAP}*}(\mathbf{Z}_1, \mathbf{Z}_2) = \frac{1}{K} \sum_k \text{Tr}(\mathbf{A}_{12,k}). \quad (10)$$

We opt for this version in our work, noting that the two measures are equivalent when $N_1 = N_2 = \dots = N_K$.

3.3.2 Text-to-instrument (T2I)

The average CLAP score $s_{\text{CLAP}*}$ is suitable for cases with a one-to-one match between ground truth prompts and their corresponding audio examples. However, it can deteriorate for T2I, where a single CLAP text embedding must be related to an ensemble of CLAP audio embeddings $\mathbf{Z}_{2,k}$. A naive adaptation involves comparing each audio embedding within the generated instrument to the same target text embedding. This amounts to creating $\mathbf{Z}_{1,k}$ by replicating the CLAP text embedding N_k times (whose resulting covariance is the "ideal TC" one in Figure 2a), and using it as input to Equation 10. Hence, we set out to *synthesize* a realistic ensemble of CLAP embeddings $\mathbf{Z}_{1,k}$ from a single CLAP text embedding $\mathbf{z}_{\text{CLAP},t} = E_t(\mathbf{t}_k)$, derived from the k th text prompt \mathbf{t}_k . Again, we accomplish this by leveraging statistics from available instrument data.

We construct $\mathbf{M}_{1,*} \in \mathbb{R}^{d_z \times d_p d_v}$ as the mean CLAP audio embeddings at each pitch/velocity pair across all instruments in our evaluation data, re-normalizing them upon

averaging. We posit that a text prompt implies a specific pitch/velocity (e.g., "softly plucked upright bass" suggests a low pitch/velocity). To estimate the corresponding pitch $\hat{\rho}_k$ and velocity $\hat{\nu}_k$ for a given prompt, and to identify its closest template $\hat{\mu}_{1,k}$, we use $\mathbf{M}_{1,*}$ as a template matching-based classifier onto $\mathbf{z}_{\text{CLAP},t}$. Accordingly, we can define

$$\mathbf{M}_{1,k} = \mathbf{M}_{1,*} + (\hat{\mu}_{1,k} - \mathbf{z}_{\text{CLAP},t}) \quad (11)$$

such that $\mathbf{M}_{1,k}$ is aligned to $\mathbf{z}_{\text{CLAP},t}$ at $\hat{\rho}_k/\hat{\nu}_k$. Re-normalizing, we have $\mathbf{Z}_{1,k} = \mathbf{M}_{1,k}/\|\mathbf{M}_{1,k}\|$. Figure 2b illustrates a covariance matrix derived from this approach for a given text prompt. This *translation* method improves upon naive replication, but contains higher cross-correlations than in $\mathbf{A}_{11,*}$. Finally, we derive a *coloration* transformation $\mathbf{Z}_{1,k} \leftarrow Y(\mathbf{Z}_{1,k}, \mathbf{A}_{11,*})$ through standard Eigendecomposition techniques, resulting in a $\mathbf{Z}_{1,k}$ with covariance $\mathbf{A}_{11,*}$, as in Figure 2c.

4. EXPERIMENTAL RESULTS

We train models on the NSynth dataset [3], pruning it according to our specified 88-key pitch range. We re-sample the 16 kHz dataset to 44.1 kHz, viewing it as a proxy in lieu of an equally comprehensive full-band alternative. Models are trained to minimize the cross-entropy \mathcal{L}_{ce} between predicted codes $\hat{\mathbf{c}}$ and ground truth \mathbf{c} , over 1M training steps with AdamW optimizer, a batch size of 48, and a cosine-annealed schedule as in [16] with an initial learning rate of 10^{-3} . We primarily analyze the impact of the proposed CLAP conditioning training variants with AR inference. Additionally, we train a baseline CLAP model with MAGNeT-style iterative decoding to compare its relative performance. To promote consistency in generated samples used for evaluation, we fix the random seed of our categorical samplers, ensuring that generations undergo the same random sampling trajectory. We refer readers to our supplementary materials available at <https://gen-inst.netlify.app/>.

We evaluate and analyze the models through several means. We liken S2I to a reconstruction of the NSynth test set [1] adapted to our inference condition, as a user can provide a sample at any pitch/velocity available to them and models must render its timbre over all pitch/velocity queries. We simulate this by randomly selecting a single query CLAP audio embedding for each instrument, using it to generate all other samples within the instrument. For

Model	Inference	FAD _{VGGish} ↓	FAD _{CLAP} ↓	s_{CLAP^*} ↑	TC _{CLAP^*} ↑
Baseline CLAP	AR	1.781	0.214	0.626	0.937
Random CLAP	AR	1.558	0.196	0.656	0.929
Fixed CLAP	AR	1.951	0.225	0.637	0.951
Baseline CLAP	MAGNeT	1.974	0.263	0.561	0.931

Table 2. Objective S2I evaluation over the NSynth test set.

Model	FAD _{VGGish} ↓	FAD _{CLAP} ↓	TC _{CLAP^*} ↑	Naive	Translation	Coloration
Baseline CLAP	3.060	0.402	0.908	0.225	0.239	0.359
Random CLAP	2.416	0.315	0.883	0.168	0.224	0.361
Fixed CLAP	3.668	0.427	0.932	0.171	0.204	0.333

Table 3. Objective T2I evaluation over a curated set of text prompts (left), and using s_{CLAP^*} ↑ comparing naive application of CLAP text embeddings against the proposed translation and coloration methods for synthesizing $Z_{1,k}$ (right).

T2I, we curate 25 text prompts of varying complexity, generating instruments accordingly.

4.1 Objective evaluation

We analyze generations across S2I and T2I, using FAD (for overall expressivity and fidelity), s_{CLAP^*} (for prompt correspondence), and TC_{CLAP^*} (for TC) to evaluate models quantitatively. To compute FAD scores for T2I, we relate generated instruments to the NSynth test set in the absence of the ground truth audio. Lastly, we compare the different s_{CLAP^*} versions for T2I introduced in Section 3.3.2.

Quantitative results for S2I and T2I are summarized in Tables 2 and 3, respectively. For S2I, the random CLAP variant outperforms other models in terms of FAD and s_{CLAP^*} at the expense of reduced TC. The converse is true for the fixed CLAP variant, which outperforms in TC. While we do not prescribe which factor is most crucial to overall instrument quality, we do assert that TC is an important element for overall playability. The baseline CLAP approach slots itself in the middle with regards to all criteria. Its MAGNeT variant exhibits degraded performance, but generates samples with $7\times$ fewer inference steps. These findings are largely mirrored in the T2I case. Interestingly, the baseline CLAP variant seemingly outperforms models in terms of s_{CLAP^*} using a naively adapted measure. The translation method increases scores across all models. Lastly, we see that the random CLAP model (marginally) outperforms other variants when using the coloration method, in line with S2I. Note that this version of the measure significantly bolsters s_{CLAP^*} across all models relative to naive replication and translation, so we argue that it is best-suited for computing T2I s_{CLAP^*} .

4.2 Subjective evaluation

We used the Multiple Stimuli with Hidden Reference and Anchor (MUSHRA) and Mean Opinion Scores (MOS) methods [30] to evaluate model variants subjectively. The MUSHRA test was catered to S2I, and involved participants rating the quality of individual samples generated by different models against a hidden reference (i.e. a ground truth sample) and an anchor (i.e. a sample generated by a

randomly initialized model). We performed a 1-5 Likert scale MOS test for T2I scenarios, where participants evaluated the audio outputs generated from text prompts based on overall playability and TC. Our accompanying website demonstrates the nature of trials used in our evaluation.

In total, 62 participants took part in our two-phase evaluation, with results summarized in Table 4. Note that most participants possess expert listening skills and have been involved in virtual instrument creation for several years, contributing to slightly lower absolute results than anticipated. Listening test results were consistent with our objective evaluation, confirming the two assertions of our work: (1) random CLAP improves expressivity over baseline CLAP by virtue of its data augmentation, and (2) fixed CLAP improves TC over baseline CLAP because its training more closely resembles the inference condition.

Model	MUSHRA	MOS
Baseline CLAP	56.08	2.290
Random CLAP	63.35	2.661
Fixed CLAP	57.96	2.820
Ground truth	98.45	–
Anchor	0.442	–

Table 4. Summary of our subjective listening tests.

5. CONCLUSIONS

In this work, we proposed methods for generating sample-based musical instruments from text or audio prompts using neural audio codec language models. We considered different CLAP conditioning variants based on the unique challenge of our task, whereby a set of samples that are timbrally consistent must be generated from a single prompt. We proposed metrics to assess sample-based instruments through various means. Extensive evaluations showcased the effectiveness of our methods, underscoring a compromise between expressivity and TC. Future work will enable deeper control for sample generation, where adapters could be used to augment a base model [31]. We would also like to improve system fidelity, scaling models to larger sizes with fine-tuned modules [9].

6. ETHICS STATEMENT

We have intentionally pursued this task as a topic for scientific research as an alternative to more conventional prompt-to-media systems. The spirit of this work is specifically to expand sound synthesis possibilities for music creators in order to realize their artistic visions. Moreover, we feel that our resulting system and its intents pose far less risk to personal attack/misrepresentation as well as the livelihood of creatives, and is less susceptible to incrimination/impersonation attempts relative to the forms of generative models that have caused increased levels of concern within the general population [32].

Beyond our primary ethical concerns, we also recognize the environmental implications of our computational practices. Our experiments were carried out using Amazon Web Services in the *us-gov-east-1* region, with a carbon efficiency of 0.57 kgCO₂eq per kilowatt-hour. One training of our model entailed approximately 96 hours of computation on Intel Xeon E5-2686 v4 (Broadwell) hardware using a single V100 GPU, culminating in an estimated total emission of 7.93 kgCO₂eq. This estimation was facilitated by the Machine Learning Impact calculator [33]. In acknowledging our environmental impact, we underscore the importance of integrating sustainability considerations into the research process, reflecting on the imperative to balance innovation with ecological responsibility.

7. REFERENCES

- [1] G. Narita, J. Shimizu, and T. Akama, “GANStrument: Adversarial Instrument Sound Synthesis with Pitch-Invariant Instance Conditioning,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Jun. 2023.
- [2] H. Chang, H. Zhang, J. Barber, A. Maschinot, J. Lezama, L. Jiang, M.-H. Yang, K. P. Murphy, W. T. Freeman, M. Rubinstein, Y. Li, and D. Krishnan, “Muse: Text-To-Image Generation via Masked Generative Transformers,” in *Proceedings of the International Conference on Machine Learning*, Jul. 2023.
- [3] J. Engel, C. Resnick, A. Roberts, S. Dieleman, D. Eck, K. Simonyan, and M. Norouzi, “Neural audio synthesis of musical notes with WaveNet autoencoders,” in *Proceedings of the International Conference on Machine Learning*, Aug. 2017.
- [4] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “WaveNet: A generative model for raw audio,” *arXiv:1609.03499*, 2016.
- [5] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, “GANSynth: Adversarial Neural Audio Synthesis,” in *Proceedings of the International Conference on Learning Representations*, May 2019.
- [6] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, “DDSP: Differentiable Digital Signal Processing,” in *Proceedings of the International Conference on Learning Representations*, April 2020.
- [7] D. Y. Wu, W. Y. Hsiao, F. R. Yang, O. Friedman, W. Jackson, S. Bruzenak, Y. W. Liu, and Y. H. Yang, “DDSP-Based Singing Vocoders: A New Subtractive Based Synthesizer and A Comprehensive Evaluation,” in *Proceedings of the International Society for Music Information Retrieval Conference*, Dec. 2022.
- [8] A. Caillon and P. Esling, “RAVE: A Variational Autoencoder for Fast and High-Quality Neural Audio Synthesis,” *arXiv:2111.05011*, Nov. 2021.
- [9] Z. Evans, C. Carr, J. Taylor, S. H. Hawley, and J. Pons, “Fast timing-conditioned latent audio diffusion,” *arXiv:2402.04825*, Feb. 2024.
- [10] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, “SoundStream: An End-to-End Neural Audio Codec,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Nov. 2021.
- [11] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, “High-fidelity audio compression with improved RVQGAN,” *Conference on Neural Information Processing Systems*, Dec. 2023.
- [12] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi, and N. Zeghidour, “AudioLM: a Language Modeling Approach to Audio Generation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Jun. 2023.
- [13] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, L. He, S. Zhao, and F. Wei, “Neural Codec Language Models are Zero-Shot Text to Speech Synthesizers,” *arXiv:2301.02111*, Jan. 2023.
- [14] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, “AudioGen: Textually Guided Audio Generation,” in *Proceedings of the International Conference on Learning Representations*, 2023.
- [15] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank, “MusicLM: Generating Music From Text,” *arXiv:2301.11325*, Jan. 2023.
- [16] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and Controllable Music Generation,” in *Proceedings of the Conference on Neural Information Processing Systems*, Dec. 2023.
- [17] J. D. Parker, J. Spijkervet, K. Kosta, F. Yesiler, B. Kuznetsov, J. C. Wang, M. Avent, J. Chen, and

- D. Le, "StemGen: A music generation model that listens," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Apr. 2024.
- [18] H. F. Garcia, P. Seetharaman, R. Kumar, and B. Pardo, "VampNet: Music generation via masked acoustic token modeling," in *Proceedings of the International Society for Music Information Retrieval Conference*, Nov. 2023.
- [19] S. Nercessian and J. Imort, "InstrumentGen: Generating sample-based musical instruments from text," in *Neural Information Processing Systems Workshop on Machine Learning for Audio*, Dec. 2023.
- [20] B. Hayes, J. Shier, G. Fazerkas, A. McPherson, and C. Saitis, "A Review of Differentiable Digital Signal Processing for Music and Speech Synthesis," *Frontiers in Signal Processing*, Jan. 2024.
- [21] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-Scale Contrastive Language-Audio Pretraining with Feature Fusion and Keyword-to-Caption Augmentation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Jun. 2023.
- [22] A. Ziv, I. Gat, G. L. Lan, T. Remez, F. Kreuk, J. Copet, A. Défossez, G. Synnaeve, and Y. Adi, "Masked audio generative modeling," in *Proceedings of the International Conference on Learning Representations*, May 2024.
- [23] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High Fidelity Neural Audio Compression," *Transactions on Machine Learning Research*, Sep. 2023.
- [24] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," in *Proceedings of the International Society for Music Information Retrieval Conference*, Sep. 2018.
- [25] V. Vapnik and R. Izmailov, "Learning using privileged information: Similarity control and knowledge transfer," *Journal of Machine Learning Research*, Nov. 2015.
- [26] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "HTS-AT: A Hierarchical Token-Semantic Audio Transformer for Sound Classification and Detection," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2022.
- [27] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv:1907.11692*, Jul. 2019.
- [28] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Frechet audio distance: A metric for evaluating music enhancement algorithms," *arXiv:1812.08466*, Dec. 2018.
- [29] A. Gui, H. Gamper, S. Braun, and D. Emmanouilidou, "Adapting Frechet audio distance for generative music evaluation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Apr. 2024.
- [30] J. Camp, T. Kenter, L. Finkelstein, and R. Clark, "MOS vs. AB: Evaluating text-to-speech systems reliably using clustered standard errors," in *Proceedings of Interspeech*, Aug. 2023.
- [31] K. Sohn, N. Ruiz, K. Lee, D. C. Chin, I. Blok, H. Chang, J. Barber, L. Jiang, G. Entis, Y. Li, Y. Hao, I. Essa, M. Rubinstein, and D. Krishnan, "StyleDrop: Text-to-Image Generation in Any Style," in *Proceedings of the Conference on Neural Information Processing Systems*, Dec. 2023.
- [32] J. Barnet, "The ethical implications of generative audio models: A systematic literature review," in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, Aug. 2023.
- [33] A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres, "Quantifying the carbon emissions of machine learning," *arXiv:1910.09700*, 2019.