Creative Text-to-Audio Generation via Synthesizer Programming

Manuel Cherep [*1] **Nikhil Singh** [*1] **Jessica Shand** [1]

Abstract

Neural audio synthesis methods now allow specifying ideas in natural language. However, these methods produce results that cannot be easily tweaked, as they are based on large latent spaces and up to billions of uninterpretable parameters. We propose a text-to-audio generation method that leverages a virtual modular sound synthesizer with only 78 parameters. Synthesizers have long been used by skilled sound designers for media like music and film due to their flexibility and intuitive controls. Our method, *CTAG*, iteratively updates a synthesizer's parameters to produce high-quality audio renderings of text prompts that can be easily inspected and tweaked. Sounds produced this way are also more abstract, capturing essential conceptual features over fine-grained acoustic details, akin to how simple sketches can vividly convey visual concepts. Our results show how *CTAG* produces sounds that are distinctive, perceived as artistic, and yet similarly identifiable to recent neural audio synthesis models, positioning it as a valuable and complementary tool.[1]

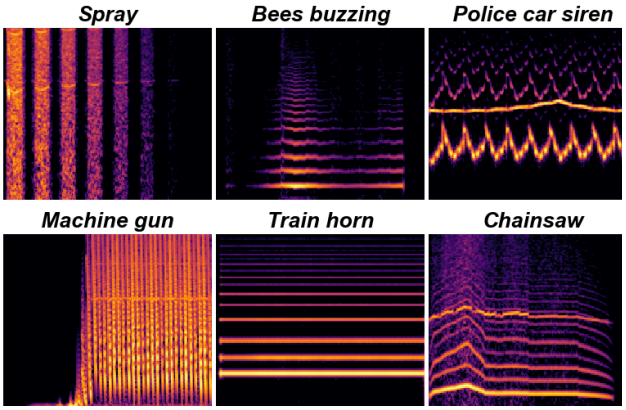

Figure 1. CTAG leverages a virtual modular synthesizer to generate sounds capturing the semantics of user-provided text prompts in a sketch-like way, rather than being acoustically literal. Spectrograms of auditory outputs corresponding to six text prompts showcase the range of sounds this approach can yield, accompanied by a fully interpretable and controllable parameter space.

1. Introduction

> *"Of course, bubbles don't make sound, but this is the magic of sound design...you can create the concept of a sound and it seems real."*
>
> — Suzanne Ciani

In creative sound design, realism isn't everything. In the late 1970s, composer Suzanne Ciani famously demonstrated this principle with her iconic *Coca Cola pop and pour* sound effect. This sound, which has become synonymous with the refreshing experience of opening a soda, was not recorded from an actual soda bottle, but skillfully crafted using a Buchla synthesizer. Ciani's work illustrates the immense power of abstraction in auditory representation, where the essence of a concept can be expressed without mimicking real-world acoustic details, while achieving greater impact.

This approach extends beyond single examples into the domain of procedural sound design: creating sounds algorithmically using parameters that can be manipulated to achieve desired sonic effects. By applying procedural techniques, sound designers can often transcend what's physically plausible to obtain by recording real-world events. These methods can lead to highly evocative and expressive sounds in music, film, video games, advertising, product design, and other media.

Neural audio synthesis methods have transformed the state of sound design, enabling specifying sound ideas using intuitive inputs like textual prompts. However, there remains unrealized potential in integrating expressive sound design principles into neural audio synthesis. Current techniques prioritize acoustic recreation and end-to-end application, often overlooking creative possibilities for evoking emotions

[*]Equal contribution [1]Media Lab, Massachusetts Institute of Technology, Cambridge MA, USA. Correspondence to: Manuel Cherep <mcherep@mit.edu>, Nikhil Singh <nsingh1@mit.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

[1]ctag.media.mit.edu

or concepts, and interactive aspects like manipulating, iterating, and interpolating between sounds. While recent advances showcase remarkable capabilities in replicating real-world sounds, this emphasis can limit the creative palette and expressive potential of generated audio. We propose a method to bridge this gap.

Overall, this work contributes:

- A novel method that integrates a virtual modular synthesizer with a pretrained audio-language model for generating sounds that resonate with human intuition without being literal representations.
- A lightweight, fully interpretable, and controllable synthesizer resulting from our approach, allowing for easy inspection and tweaking for creative purposes.
- Extensive experiments evaluating different approaches to solving this problem, varying optimization algorithms, sound durations, and synthesis architectures.
- Qualitative and quantitative results that highlight how sounds from our method have distinct features from those produced by other neural audio generators, while still being identified at similar rates. We conduct a user study as a gold standard evaluation, given the novelty of the task, which shows the identifiability and potential artistic value of *CTAG*'s sounds.
- Examples of this approach generating several datasets of sounds with their synthesis parameters, and interpolating between different sounds in the parameter space.

We will open-source our approach, both to provide a tool for novices and experts alike to realize their ideas, as well as to provoke future audio generation paradigms that recognize abstraction as an important factor for creative expression.

2. Related Work

2.1. Sound Synthesis

Neural audio synthesis consists of two main strands: approaches that generate audio waveforms directly in the time domain, and those that do so in the frequency domain. WaveNet (Oord et al., 2016) notably introduced an autoregressive approach to audio synthesis by predicting one sample at a time. This slow iterative sampling approach, later refined in SampleRNN (Mehri et al., 2016) and WaveRNN (Kalchbrenner et al., 2018), reflects the sequential nature of audio data, in contrast to images wherein GANs with global latent conditioning and efficient parallel sampling quickly became a dominant method for synthesis. Later, WaveGAN (Donahue et al., 2018) and GANSynth (Engel et al., 2019) demonstrated that GANs could in fact be used to synthesize locally-coherent audio, outperforming sequential models' speed by several orders of magnitude while maintaining a focus on high-fidelity, natural-sounding audio.

A third strand of so-called *oscillator* models, largely propelled by Differentiable Digital Signal Processing (DDSP) (Engel et al., 2020) is physically and perceptually motivated by the rich history of synthesis and signal processing techniques. Our approach is motivated by this direction, but relies on a simple synthesizer architecture, CLAP (Wu et al., 2023), for text-conditioning, and gradient-free optimization to provide a simple, training-free solution.

2.2. Language-Sound Correspondence

Advances in multi-modal sound-language models have been partly motivated by CLIP (Radford et al., 2021) for images. Wav2CLIP (Wu et al., 2022) builds directly onto CLIP by adding an audio encoder, and VQGAN+CLIP (Crowson et al., 2022) generates and edits images guided by text prompts. Audio representation models, such as Microsoft's CLAP (Elizalde et al., 2023) and LAION-CLAP (Wu et al., 2023), emulate CLIP's approach by using contrastive learning on audio-text pairs. We use LAION-CLAP as our audio-language model in this work.

Other recent approaches cast audio generation as a language modeling task. AudioGen (Kreuk et al., 2022) is an autoregressive model conditioned on text inputs. AudioLM (Borsos et al., 2023) uses a multi-stage Transformer-based language model. WavJourney (Liu et al., 2023b) uses text instructions to create scripts, which are then used for compositional audio creation. Make-An-Audio 1 and 2 (Huang et al., 2023b;a) offer text-to-audio synthesis with prompt-enhanced diffusion models, using CLAP to map text to latent representations with a spectrogram autoencoder. AudioLDM (Liu et al., 2023a) learns continuous audio representations from CLAP latents and can perform text-guided audio manipulations. We compare to two state-of-the-art solutions, namely *AudioGen* and *AudioLDM*, in our experiments. Our goals differ significantly from those of these models, as we seek to generate abstract yet high-quality sounds, rather than literal recording-like renditions.

2.3. Abstract Synthesis

Visual sketching offers an intuitive analog to abstract sound synthesis. Minimal representations like monochromatic line drawings might use only straight lines and curves with no additional shading or color. These renderings are non-photorealistic; they evocatively convey meaning while emphasizing a subject's essence over its real-world presentation. They can also reveal insights about a subject's underlying geometry, proportions, and symbolism that may be obscured in more realistic depictions.

The problem of computing recognizable and insightful abstract renderings has seen more progress in the visual than the audio domain. CLIPasso (Vinker et al., 2022) leverages CLIP to distill semantic meanings from images and sketches alike and thereby guide text-to-image generation, varying the number of strokes according to the desired level of abstraction. CLIPTexture (Song, 2022) enables a user to manipulate a simple sketch or layout through textual descriptions. CLIPVG (Song et al., 2023) follows the same progressive optimization approach, but performs image manipulation using vector graphics rather than pixels. ES-CLIP (Tian & Ha, 2022) tackles the problem via evolution strategies, generating configurations of colored triangles on a canvas, then assessing their fitness for further iteration. We were inspired by this approach, though we rely on the well-established, easily interpretable, and tweakable paradigm of modular synthesis.

In the auditory domain, the Sound Sketchpad (Singh, 2021) combines sounds together using audio-visual sketches, and the SkAT-VG project (Rocchesso et al., 2015) applies vocal and gestural manipulation as natural sketching tools. In our approach, we focus on language input, and synthesis rather than the composition of existing sounds.

2.4. Interpretable and Controllable Synthesis

Interpretability and controllability of results is essential to human-machine co-creation, in which it is often desirable to closely examine, understand, and fine-tune an artifact. For creative sound design using neural synthesis methods, it can be impossible to retrace decisions made by a complex neural synthesis model en route to synthesizing an output. The model may also not provide any opportunity to iteratively refine the output. Some prior work (Young et al., 2022) highlights the potential of program synthesis for interpretability in sequence data, including music. Some neural synthesis models integrate techniques like timbre-regularization (Esling et al., 2018) to bridge powerful synthesis methods with perceptually-motivated organization of latent spaces. By contrast, our approach offers a fully interpretable and controllable parameter space without requiring us to develop additional neural infrastructure.

2.5. The Synthesizer Programming Problem

Despite the near-ubiquitous presence of synthesized sound in modern music, synthesizer programming—that is, the act of creating new sounds through careful analysis and modulation of synthesizer parameters—is a complex task that can often impede the creative process, if not bar entry entirely. In particular, the conceptual disconnect between parameter settings and the associated auditory output (Shier, 2021) makes synthesizer programming especially non-intuitive without special training. Recent work has investigated techniques for inverse synthesis—given a target sound, infer the parameter setting that will emulate the sound to the closest extent possible—on both musical sounds (Yee-King et al., 2018) and real-world sounds, such as animal vocalizations (Hagiwara et al., 2022), including deep learning methods to learn invertible mappings (Esling et al., 2020). However, this task still requires a specific audio clip to start. We provide text-to-parameter inference to bridge this gap, generalizing beyond specific audio files to broader semantic notions of arbitrary sounds.

3. Methods

Our methodology hinges on three pillars: a synthesizer, implemented via SYNTHAX (Cherep & Singh, 2023), gradient-free optimization methods, implemented via the Evosax (Lange, 2023) evolutionary optimization library, and an objective function based on the LAION-CLAP (Wu et al., 2023) model, which we use to estimate semantic alignment between the synthesized audio and its corresponding text prompt (see Figure 2 for an overview of the pipeline).

3.1. Synthesizer

We use a simple synthesizer implementation available in SYNTHAX, a fast modular synthesizer written in JAX (Bradbury et al., 2018). We specifically use the *Voice* synthesizer architecture, adapted from *torchsynth* (Turian et al., 2021), which has already been used for programmatic resynthesis of sounds (Hagiwara et al., 2022). It consists of 78 parameters for a monophonic keyboard, two low-frequency oscillators (LFOs), six ADSR envelopes, a sine voltage-controlled oscillator (VCO), a square-saw VCO, a noise generator, voltage-controlled amplifiers (VCAs), a modulation mixer and an audio mixer. All parameters are initialized uniformly, $\theta_i \sim U(0, 1)$.

In addition to this architecture, we evaluate the following variants in increasing order of architectural complexity:

- *ShapedNoise*: An 18 parameter synthesizer consisting of a noise generator, and two control elements to shape the noise amplitude over time: an ADSR envelope, and a low-frequency oscillator (LFO). These are combined into a modulation signal through a modulation matrix, which itself has learnable weights for this combination.
- *OneOsc*: A 23 parameter synthesizer consisting of a sine wave voltage-controlled oscillator (VCO), and the same two control elements as above. These elements are combined into two signals through a modulation matrix, one each for frequency and amplitude.
- *NoLFO*: A 29 parameter two-VCO synthesizer, where one is a sine wave oscillator and the other is a square-saw wave oscillator with a "shape" parameter which

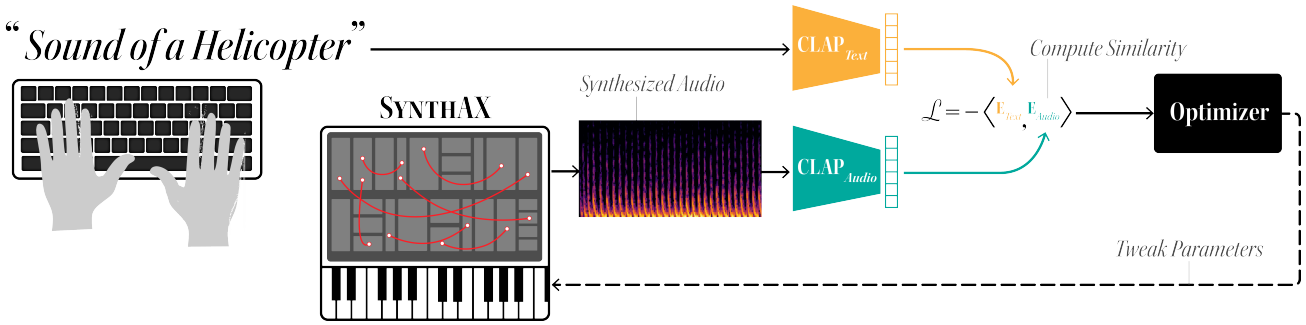

Figure 2. High-level overview: we use the LAION-CLAP model (Wu et al., 2023) to compute the similarity between a user-provided text prompt and SYNTHAX's (Cherep & Singh, 2023) output. The optimization procedure iteratively adjusts the parameter settings.

controls the degree of "square-ness" vs. "saw-ness". This synthesizer has no LFO components, all modulation is conducted by two ADSR envelopes combined into four separate modulation signals (pitch and amplitude controls for each of the two VCOs).

- *NoNoise*: A 51 parameter synthesizer with two VCOs (as before), and a more complex modulation structure. Here, there is a single LFO, but there are additional ADSRs to modulate the frequency and amplitude of this LFO. The modulated LFO and two ADSR envelopes comprise the inputs to the modulation matrix.

- *Voice+FM*: A 130 parameter synthesizer which adds a frequency modulation (FM) component to the original *Voice* architecture.

For reference, an ADSR envelope is a piecewise control signal consisting of linear or exponential segments: **A**ttack, **D**ecay, and **R**elease, which specify the duration of each envelope segment. The **S**ustain parameter is the level of the control signal after the decay phase. An LFO is an oscillator whose frequencies are typically lower than audible frequencies, i.e. below 20-40 Hz. These are used for periodic control of synthesis parameters.

In all our experiments, the synthesizer has a control rate of 480 Hz and the audio is generated in batches at a sample rate of 48 kHz. This sample rate is much higher than that commonly used for neural audio synthesis systems (often 16 kHz) and therefore admits much more high-frequency content to be generated.

3.2. Optimization

During initial experiments, we found the gradients of our differentiable synthesizer to be highly unstable. This instability hindered optimization performance even after attempting mitigation strategies. Recent works in abstract visual synthesis have shown that non-gradient methods can achieve state-of-the-art results without relying on gradient information (Tian & Ha, 2022). Given these findings, we decided to explore non-gradient approaches which are more suitable for our synthesizer's instability and have demonstrated effectiveness for this task. Focusing efforts here allowed us to sidestep gradient issues while leveraging successful techniques from related synthesis domains.

We experimented with several non-gradient optimization algorithms, using implementations from Evosax (Lange, 2023). Specifically, we examined simple baselines like

Algorithm 1 Our optimization procedure for producing sounds in *CTAG*. Note: d is the number of parameters of the synthesizer S; for simplicity we omit batches.

Require: Text prompt p
Require: Population/batch size N
Require: Iterations M

Components:
CLAP text embedding model $C_t(p) \rightarrow E^p$
SynthAX synthesizer $S(\Theta) \rightarrow X^a$
CLAP audio embedding model $C_a(X^a) \rightarrow E^{X^a}$
Optimization Strategy: O

Initialize:
Synthesis parameters $\Theta = \{\theta_1, \ldots, \theta_N\}, \theta_i \sim U(0,1)$
Flattened parameters $\Theta_f \in [0,1]^{N \times d} = Flatten(\Theta)$

for $i = 1$ **to** M **do**
$\Theta_{f_{\text{new}}} \leftarrow O_{\text{ask}}(\Theta)$ *Generate candidates*
$\Theta_{\text{new}} \leftarrow \text{Reshape}(\Theta_{f_{\text{new}}})$ *Reshape*
$X^a \leftarrow S(\Theta_{\text{new}})$ *Synthesize audio*
$E^{X^a} \leftarrow C_a(X^a)$ *Get audio embeddings*
$F \leftarrow -E^{X^a} E^{p^T}$ *Compute fitness*
$O_{\text{tell}}(\Theta_{\text{new}}, F)$ *Update optimizer state*
$\Theta \leftarrow \Theta_{\text{new}}$
end for

$\theta^* = \arg\min_\theta F$ *Select optimal parameters*

random search and a simple genetic algorithm (Such et al., 2017), well-known methods like CMA-ES (Hansen & Ostermeier, 2001) and Particle Swarm Optimization (Kennedy & Eberhart, 1995), and state-of-the-art methods like Learned and Discovered Evolution Strategies (Lange et al., 2023). For each algorithm, we first tuned hyperparameters using Bayesian optimization via the Adaptive Experimentation (AX) platform (Bakshy et al., 2018). We tuned for 50 trials on the ESC-10 dataset, a subset of ESC-50 (Piczak, 2015). Note that the hyperparameter tuning uses no privileged information and can easily be applied downstream on new prompt sets to maximize the performance.

The optimization procedure is specified in Algorithm 1.

3.3. Objective Function

We use LAION-CLAP (Wu et al., 2023) with an HTSAT-based audio encoder (Chen et al., 2022) and a RoBERTa-based text encoder (Liu et al., 2019). We used the *audioset-best* checkpoint for general audio less than 10 seconds long.

The encoders process the audio data X_i^a in batches of size $\mathcal{B}$ where $\mathcal{B}$ corresponds to the optimizer's population size, along with a prompt p. Note that (X_i^a, p) is one particular pair of synthesized audio with input text prompt. We extract the audio embeddings $E_{\mathcal{B}}^a \in \mathbb{R}^{\mathcal{B}\times 512}$ and the text embeddings $E^p \in \mathbb{R}^{1\times 512}$ with the encoders and use them to calculate the similarity score between a batch of audio data and a specific prompt.

$$X_i^a = \mathcal{S}(\theta_i) \tag{1}$$

$$\theta^* = \arg\min_{\theta} \; -E_i^{\mathcal{S}(\theta_i)} E^{pT} \tag{2}$$

Equation (1) shows how the synthesizer $\mathcal{S}$ takes parameters θ_i and produces a sound (in practice, this is done batched). Then Equation (2) formulates the optimization problem to optimize the similarity score between each audio in the batch and one given text prompt using their corresponding embeddings.

3.4. Evaluation Metrics

Since we propose a novel synthesis task without existing evaluation metrics, we devise a principled evaluation suite that allows us to quantitatively assess our contributions, in addition to qualitatively reviewing synthesized examples.

Classification Experiments To determine whether our generated sounds are more abstract than neural synthesis methods, we compared results on pretrained classifiers with sounds generated from their class labels. Lower scores can indicate a distribution shift from real audio, despite explicitly optimizing for similarity to the label. We complement with human listener ratings.

Without a perfect synthesis engine, any methods to generate sound will introduce a distribution shift from real audio. In our case, there is a deliberate domain shift to abstract audio. We evaluate on two well-known datasets. The first is ESC-50, a 50-class canonical environmental sound classification dataset (Piczak, 2015). The second is a subset of AudioSet (Gemmeke et al., 2017); the full ontology of classes is very large (over 500). We consider classes from "sounds of things" given that this category contains the most sub-classes and sub-selected the top 50 classes by number of annotations, removing duplicates or equivalent classes. We use a pretrained Audio Spectrogram Transformer (AST) model for AudioSet-50, and fine-tune an AST for ESC-50 classification (Gong et al., 2021). When evaluating on AudioSet-50, we mask the remaining logits to effectively make it a 50-class classifier.

Synthesis Quality A significant benefit of our approach is synthesizing clean audio using signal generators while keeping attributes like sample rate flexible. We find synthesized sounds also often exaggerate aspects of the prompts, resulting in large variations in acoustic properties over time. Evaluating audio quality reference-free is challenging, so we examine acoustic features that correlate with these aspects (such as high-frequency content and spectral variation).

User Study We conduct a listening test with human evaluators. We ask them to classify sounds, rate their confidence, and rate sounds along a scale from realistic portrayal to artistic interpretation. This offers us the most direct signal of our abstraction-related goal. We share details on this study in the next subsection. We compared against the recent neural generation methods *AudioLDM* (Liu et al., 2023a) and *AudioGen* (Kreuk et al., 2022).

From our 50-prompt subset of AudioSet (Gemmeke et al., 2017) classes, we randomly selected 10 for this study. We used text embeddings of the labels with a facility location submodular optimization algorithm from the apricot package (Schreiber et al., 2020) to select a modest-sized semantically representative subset. Within each prompt, we randomly sampled two of 10 available *CTAG* sounds. The prompts were: *Truck air brake*, *Water tap*, *Train horn*, *Motorcycle*, *Microwave oven*, *Liquid slosh*, *Chainsaw*, *Airplane*, *Bicycle bell*, and *Machine gun*. For *AudioLDM* and *AudioGen*, we used their default parameters to generate two sounds per prompt.

This study was determined to be exempt by our institution's IRB. Each participant rated 60 sounds (20 per method) in random order. To examine category-level recognition, par-

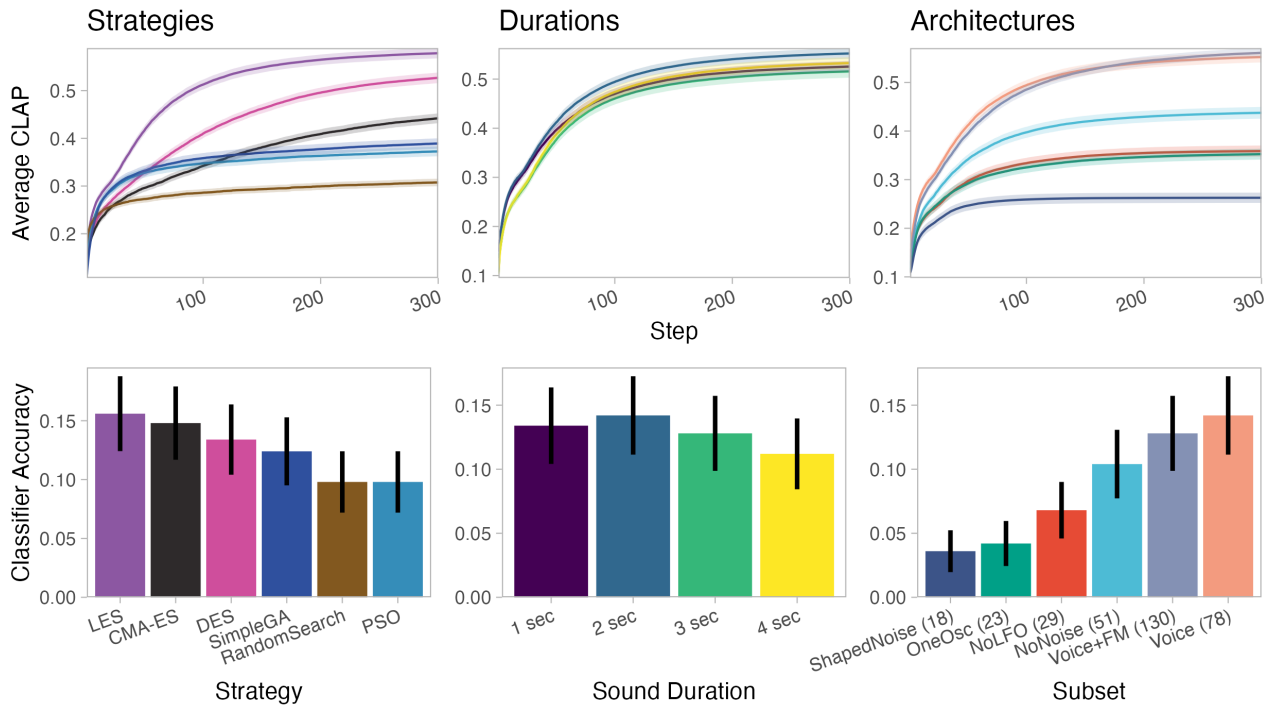

Figure 3. Results from our ablation study; all experiments are conducted with ESC-50. **(Top)** CLAP maximization curves, averaged across 10 iterations for each of the 50 prompts. Colored bands show 95% confidence intervals. **(Bottom)** Classification accuracy, with error bars showing 95% confidence intervals. Top and bottom plots share colors. **(Left)** Performance of different algorithms, with hyperparameters tuned on ESC-10. LES is strongest in both optimization and downstream classification. **(Center)** Different sound durations; we find 2 seconds to be strongest. **(Right)** Impact of synthesizer architecture, finding strongest results from the *Voice* model. Parameter counts are given in parenthesis, such as (78) for *Voice*.

ticipants were asked to select a category given a list of options and rate their confidence. To determine whether our generated sounds were perceived as (abstract) artistic interpretations, we posed the question: "Would you associate this sound more with a realistic portrayal or an artistic interpretation of the label that you selected?" with options on a scale from 1 (realistic portrayal) to 5 (artistic interpretation). We modeled participant responses with mixed-effects logistic and linear regression models and post-hoc contrasts.

4. Results

4.1. Ablation Studies

Figure 3 shows results from our ablation studies, including, from left to right, (1) optimization algorithms with tuned hyperparameters, (2) sound durations, and (3) synthesis architectures. Overall, we observe that the LES algorithm significantly outperforms our other options within the computation budget of 300 iterations (more than needed for several prompts). This experiment was conducted with 2-second long sounds, which we observe in the *Durations* experiment to yield a higher overall CLAP score and classification accuracy than 1, 3, or 4-second long generations. Finally, we see that the *Voice* architecture yields the best results, offering a balance of flexibility in its parameters and modular structure, as well as ease of optimization. However, we note that expanding to larger architectures like VoiceFM could be useful for future work to explore, with more work on the optimization strategy to obtain the best results.

Based on these results, we conduct all additional experiments discussed with the LES optimizer, 2-second sounds, and the *Voice* architecture. We conducted a full hyperparameter tuning run with 50 trials of all ESC-50 prompts to obtain the final optimization hyperparameters.

4.2. Qualitative Results

4.2.1. Examples

Figure 1 shows spectrograms of sounds—given in the supplementary material—corresponding to six text prompts. The "spray" shows bands of noisy bursts, reflecting the short, sharp sound of aerosol being expelled. The "bees

buzzing" presents a band of low to high frequencies, encapsulating the vibrant hum of a bee. The "police car siren" is characterized by high-frequency oscillations that sharply rise and fall. The "machine gun" reveals rapid, staccato bursts of energy across a broad frequency range. The "train horn" displays horizontal bands across mid to high frequencies, illustrating the horn's fundamental tone and its partials. Lastly, the "chainsaw" spectrogram is dominated by intense, continuous mid-range frequencies, punctuated by peaks corresponding to the engine's roaring and cutting action.

4.2.2. Interpolation

In sound synthesis, interpretable parameters offer a unique opportunity for deeper insight. Our method provides a fixed set of parameters that possess this property—a salient distinction from contemporary models equipped with high-dimensional latent spaces. This interpretability extends to interpolation between parameters of distinct sounds, granting auditory access to intermediate acoustical transitions. In Figure 4, we present a systematic series of spectrograms between pairs of prompts: (1) "Spray" to "Machine gun", (2) "Train horn" to "Chainsaw", and (3) "Train wagon" to "Engine revving," with three intermediary steps linearly interpolated. This discernible gradation corroborates the capacity of our parameter space to retain congruence.

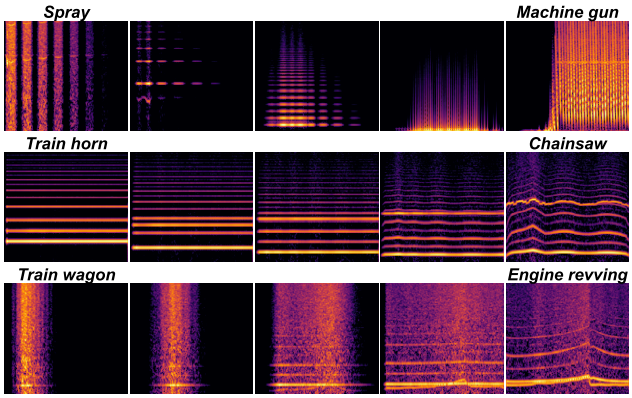

Figure 4. Spectrogram series as the result of linear interpolation of the synthesizer's parameters (1) from "Spray" (left) to "Machine gun" (right), (2) from "Train horn" to "Chainsaw", and (3) from "Train wagon" to "Engine revving". Each spectrogram in the sequence represents a step in the interpolation, highlighting the systematic shift in acoustic properties.

4.3. Classification Results

Results are shown in Table 2. On AudioSet-50, our results are higher than *AudioLDM*. On ESC-50, the classifier recognizes *CTAG*'s sounds the least, showcasing the distribution shift from its training on realistic sounds. We experimented with constructing concise and descriptive prompts from each sound class from both ESC-50 and AudioSet-50. We used GPT-4 (OpenAI, 2023) to automatically produce caption-style prompts. We also tried a simple template (i.e. "Sound of a/an ...") to compare. Table 2 also shows results for these template (*CTAG+T*) and caption-style prompts (*CTAG+C*). Introducing such strategies does not appear to greatly influence classifier identification. However, in a few cases, we observed the elaborated prompts helped to produce qualitatively more accurate results. Overall, *CTAG* sounds are classified correctly significantly higher than chance, and competitively with *AudioLDM*.

4.4. Synthesis Quality and Variation

Evaluating the quality of generated examples is challenging for two reasons. First, we lack auditory references to compare against, as we generate from text directly and never use text-audio reference pairs. Most audio quality metrics are reference-based. Second, distance-based metrics such as FAD will likely be confounded by realism. *CTAG*'s sounds are high-quality in that they can be generated at high sample rates and are free of noise or artifacts owing to real-world recording environments or neural synthesis.

To evaluate, we use auditory descriptors (implemented using Essentia (Bogdanov et al., 2013)) that are plausible correlates of these notions of quality, shown in Table 1. Spectral complexity highlights the presence of more peaks, signaling diversity in the timbral components, while flux shows greater variation of timbre over time for *CTAG* compared with other methods. Following these, HFC (high-frequency content), spectral rolloff, and spectral centroid provide signals of "brightness" or high-frequency presence in the sounds. All of these results show our method's ability to introduce high-frequency content into generated sounds, likely in part due to the higher sample rate we use.

We also report compression ratio, under variable bit rate (VBR) MP3 compression (quality = 4). Interestingly, *CTAG* achieves a higher average compression ratio. VBR generally works by applying lower ratios to more perceptually complex input. Whether related to high-frequency content or other factors, this suggests *CTAG* sounds contain more perceptual redundancy or are perceptually "simpler".

Note that none of these measures are validated as perceptual metrics of audio quality, and we do not intend to use them as such. Rather, they help us to quantify the qualitative differences we observe between *CTAG*-synthesized sounds and other text-to-audio generation models' results.

4.5. User Study

We recruited 10 participants via Prolific at $12/h for a total of $53.33, resulting in a total of 600 observations per outcome variable (i.e. accuracy, confidence, and artistic

	AudioSet-50			ESC-50		
	AudioGen	*AudioLDM*	*CTAG*	*AudioGen*	*AudioLDM*	*CTAG*
Complexity	16.50	17.65	18.06	9.60	12.94	17.76
Flux	0.08	0.11	0.18	0.06	0.09	0.15
HFC	53.25	152.06	427.03	34.49	101.32	380.74
Rolloff	2,487.71	1,628.55	7,031.67	2,254.98	1,647.51	6,996.19
Centroid	1,629.95	1,096.16	4,139.99	1,512.55	1,108.42	4,227.08
Compression Ratio	6.42	7.09	9.51	6.46	7.58	9.57

Table 1. Comparison of spectral descriptors—complexity, flux, HFC, rolloff, centroid—and audio compression ratio, across ESC-50 and AudioSet-50. Results are grouped by the evaluation of three methods: *AudioGen*, *AudioLDM*, and our method, *CTAG*.

	AudioSet-50					ESC-50				
	AudioGen	*AudioLDM*	*CTAG*	*CTAG+T*	*CTAG+C*	*AudioGen*	*AudioLDM*	*CTAG*	*CTAG+T*	*CTAG+C*
Acc (Top-1)	51.6	17.4	26.2	25.2	23.6	54.0	23.0	16.4	11.4	13.8
Acc (Top-5)	77.4	44.2	45.2	52.2	51.6	71.8	49.4	30.4	26.4	31.0

Table 2. Top-1 and Top-5 classification accuracies (%) for pre-trained classifiers with AudioSet-50 and ESC-50. We evaluated both models on results collected using *AudioGen*, *AudioLDM*, and our method with just the class labels (*CTAG*), a simple template (i.e. "Sound of a ...") for each sound (*CTAG+T*) and finally using an LLM for prompt engineering (*CTAG+C*).

interpretiveness). Table 3 contains the results, which show that our sounds were identified by listeners substantially more accurately than those from *AudioLDM* (odds ratio = 2.72, 95% CI [1.61, 4.58], $p < .0001$), and only slightly less than *AudioGen* on average (odds ratio = 0.85, 95% CI [0.51, 1.42], $p = 1$). Interestingly, though the confidence ratings replicate the ordering of the accuracy results, respondents were significantly more confident rating *AudioGen* sounds, and reported similar, lower confidence levels for both *CTAG* and *AudioLDM*. This underscores the abstractness of *CTAG*'s sounds; despite being identified more correctly, they still create uncertainty.

	AudioGen	*AudioLDM*	*CTAG*
Accuracy	59.5	34.0	56.0
Confidence	3.48	2.95	2.99
Artistic Interpretation	2.32	2.90	3.54

Table 3. User study results for sounds from *AudioGen*, *AudioLDM*, and our method, *CTAG*. We report accuracy percentage and confidence (1–5) on label identification, and average rating of the artistic interpretiveness (1–5) of the sound. Overall, *CTAG* retains competitive identifiability while being perceived as more artistic.

Results also show *CTAG* sounds were perceived to be significantly more artistically interpretive than both *AudioGen* (contrast = 1.22, 95% CI [0.93, 1.51], $t(579) = 10.20$, $p < .0001$) and *AudioLDM* (contrast = 0.65, 95% CI [0.36, 0.93], $t(579) = 5.39$, $p < .0001$).

These findings highlight our approach's benefits in capturing artistic interpretation compared to both the existing approaches. All p-values are Bonferroni-adjusted. Full results for post-hoc contrasts are available in the Appendix.

4.6. Additional Analyses

In Appendix A we provide additional analyses relating to generation time, CLAP scores, prompting strategies for the baseline models, user study results, and a visualization of the parameter space of *CTAG*-generated sounds.

5. Limitations

Our method requires iterating for each prompt from random initialization, but techniques like semantic caching to initialize to similar prompts' parameters, predictive methods for prompt-to-parameter derivation, and a user interface extension for tweaking parameters are all potential extensions to make our method more useful in real-world settings. We also focus on brief, non-mixture sounds as these are what the synthesizer is suited to modeling. Future work could explore strategies to extend the duration and complexity of sounds that can be synthesized this way.

6. Conclusion

In this work, we proposed a method for text-to-audio generation that offers a fresh perspective on neural audio synthesis by using a virtual modular synthesizer. This approach emphasizes the meaningful abstraction of auditory phenomena, contrary to prevalent methods that prioritize acoustic realism. Our results position this approach as a distinctive tool in the field of audio synthesis, capable of both expanding the toolkit of novices and experts, and stimulating new directions in audio generation research.

Acknowledgements

The authors acknowledge the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing resources that have contributed to the research results reported within this paper. Manuel was supported by a US-Spain Fulbright grant. We extend our heartfelt thanks to all participants in the user study. We also thank our meta-reviewer and reviewers, as well as Yingtao Tian, Ashvala Vinay, and Hugo Flores García for supportive comments.

Impact Statement

This work introduces a novel method for generating abstract and creative sounds from text prompts, intending to expand the creative possibilities for text-to-audio generation. We foresee several potentially positive societal impacts: (1) democratizing access to creative sound design tools, (2) stimulating new directions in audio machine learning research, (3) personalization and customization, (4) lowered likelihood of re-generating training data, and (5) lowering the computation barrier.

We do not foresee direct negative societal consequences from this contribution. However, as with any generative technology, there exists potential for misuse which should be monitored. For example, synthesized sounds are not always identifiable, and should not be used in high-stakes circumstances where identification is essential. Additionally, synthesizers can simplify complex real-world phenomena; we recognize sounds can convey a rich variety of information beyond this.

References

Bakshy, E., Dworkin, L., Karrer, B., Kashin, K., Letham, B., Murthy, A., and Singh, S. Ae: A domain-agnostic platform for adaptive experimentation. In *Conference on neural information processing systems*, pp. 1–8, 2018.

Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J. R., and Serra, X. Essentia: An audio analysis library for music information retrieval. In *International Society for Music Information Retrieval Conference*, 2013. URL https://api.semanticscholar.org/CorpusID:11200511.

Borsos, Z., Marinier, R., Vincent, D., Kharitonov, E., Pietquin, O., Sharifi, M., Roblek, D., Teboul, O., Grangier, D., Tagliasacchi, M., et al. Audiolm: a language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., et al. Jax: composable transformations of python+ numpy programs, 2018.

Chen, K., Du, X., Zhu, B., Ma, Z., Berg-Kirkpatrick, T., and Dubnov, S. Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 646–650. IEEE, 2022.

Cherep, M. and Singh, N. Synthax: A fast modular synthesizer in jax. In *Audio Engineering Society Convention 155*. Audio Engineering Society, 2023.

Crowson, K., Biderman, S., Kornis, D., Stander, D., Hallahan, E., Castricato, L., and Raff, E. Vqgan-clip: Open domain image generation and editing with natural language guidance. In *European Conference on Computer Vision*, pp. 88–105. Springer, 2022.

Donahue, C., McAuley, J., and Puckette, M. Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208*, 2018.

Elizalde, B., Deshmukh, S., Al Ismail, M., and Wang, H. Clap: Learning audio concepts from natural language supervision. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Engel, J., Agrawal, K. K., Chen, S., Gulrajani, I., Donahue, C., and Roberts, A. Gansynth: Adversarial neural audio synthesis. *arXiv preprint arXiv:1902.08710*, 2019.

Engel, J., Hantrakul, L., Gu, C., and Roberts, A. Ddsp: Differentiable digital signal processing. *arXiv preprint arXiv:2001.04643*, 2020.

Esling, P., Chemla-Romeu-Santos, A., and Bitton, A. Bridging audio analysis, perception and synthesis with perceptually-regularized variational timbre spaces. In *International Society for Music Information Retrieval Conference*, 2018. URL https://api.semanticscholar.org/CorpusID:53873046.

Esling, P., Masuda, N., and Chemla-Romeu-Santos, A. Flowsynth: Simplifying complex audio generation through explorable latent spaces with normalizing flows. In *International Joint Conference on Artificial Intelligence*, 2020. URL https://api.semanticscholar.org/CorpusID:220484250.

Gemmeke, J. F., Ellis, D. P. W., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., and Ritter, M. Audio set: An ontology and human-labeled dataset for audio events. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 776–780, 2017. URL https://api.semanticscholar.org/CorpusID:21519176.

Gong, Y., Chung, Y.-A., and Glass, J. R. Ast: Audio spectrogram transformer. *ArXiv*, abs/2104.01778, 2021. URL https://api.semanticscholar.org/CorpusID:233024831.

Hagiwara, M., Cusimano, M., and Liu, J.-Y. Modeling animal vocalizations through synthesizers. *arXiv preprint arXiv:2210.10857*, 2022.

Hansen, N. and Ostermeier, A. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.

Huang, J., Ren, Y., Huang, R., Yang, D., Ye, Z., Zhang, C., Liu, J., Yin, X., Ma, Z., and Zhao, Z. Make-an-audio 2: Temporal-enhanced text-to-audio generation. *arXiv preprint arXiv:2305.18474*, 2023a.

Huang, R., Huang, J., Yang, D., Ren, Y., Liu, L., Li, M., Ye, Z., Liu, J., Yin, X., and Zhao, Z. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. *arXiv preprint arXiv:2301.12661*, 2023b.

Kalchbrenner, N., Elsen, E., Simonyan, K., Noury, S., Casagrande, N., Lockhart, E., Stimberg, F., Oord, A., Dieleman, S., and Kavukcuoglu, K. Efficient neural audio synthesis. In *International Conference on Machine Learning*, pp. 2410–2419. PMLR, 2018.

Kennedy, J. and Eberhart, R. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pp. 1942–1948. IEEE, 1995.

Kreuk, F., Synnaeve, G., Polyak, A., Singer, U., Défossez, A., Copet, J., Parikh, D., Taigman, Y., and Adi, Y. Audiogen: Textually guided audio generation. *arXiv preprint arXiv:2209.15352*, 2022.

Lange, R., Schaul, T., Chen, Y., Lu, C., Zahavy, T., Dalibard, V., and Flennerhag, S. Discovering attention-based genetic algorithms via meta-black-box optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 929–937, 2023.

Lange, R. T. evosax: Jax-based evolution strategies. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, pp. 659–662, 2023.

Liu, H., Chen, Z., Yuan, Y., Mei, X., Liu, X., Mandic, D., Wang, W., and Plumbley, M. D. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023a.

Liu, X., Zhu, Z., Liu, H., Yuan, Y., Cui, M., Huang, Q., Liang, J., Cao, Y., Kong, Q., Plumbley, M. D., et al. Wavjourney: Compositional audio creation with large language models. *arXiv preprint arXiv:2307.14335*, 2023b.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

McInnes, L., Healy, J., and Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Mehri, S., Kumar, K., Gulrajani, I., Kumar, R., Jain, S., Sotelo, J., Courville, A., and Bengio, Y. Samplernn: An unconditional end-to-end neural audio generation model. *arXiv preprint arXiv:1612.07837*, 2016.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023.

Piczak, K. J. Esc: Dataset for environmental sound classification. *Proceedings of the 23rd ACM international conference on Multimedia*, 2015. URL https://api.semanticscholar.org/CorpusID:17567398.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Rocchesso, D., Lemaitre, G., Susini, P., Ternström, S., and Boussard, P. Sketching sound with voice and gesture. *interactions*, 22(1):38–41, 2015.

Schreiber, J., Bilmes, J., and Noble, W. S. apricot: Submodular selection for data summarization in python. *The Journal of Machine Learning Research*, 21(1):6474–6479, 2020.

Shier, J. The synthesizer programming problem: improving the usability of sound synthesizers, 2021.

Singh, N. The sound sketchpad: Expressively combining large and diverse audio collections. In *26th International Conference on Intelligent User Interfaces*, pp. 297–301, 2021.

Song, Y. Cliptexture: Text-driven texture synthesis. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 5468–5476, 2022.

Song, Y., Shao, X., Chen, K., Zhang, W., Jing, Z., and Li, M. Clipvg: Text-guided image manipulation using differentiable vector graphics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 2312–2320, 2023.

Such, F. P., Madhavan, V., Conti, E., Lehman, J., Stanley, K. O., and Clune, J. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *arXiv preprint arXiv:1712.06567*, 2017.

Tian, Y. and Ha, D. Modern evolution strategies for creativity: Fitting concrete images and abstract concepts. In *International conference on computational intelligence in music, sound, art and design (part of evostar)*, pp. 275–291. Springer, 2022.

Turian, J., Shier, J., Tzanetakis, G., McNally, K., and Henry, M. One billion audio sounds from gpu-enabled modular synthesis. In *2021 24th International Conference on Digital Audio Effects (DAFx)*, pp. 222–229. IEEE, 2021.

Vinker, Y., Pajouheshgar, E., Bo, J. Y., Bachmann, R. C., Bermano, A. H., Cohen-Or, D., Zamir, A., and Shamir, A. Clipasso: Semantically-aware object sketching. *ACM Transactions on Graphics (TOG)*, 41(4):1–11, 2022.

Wu, H.-H., Seetharaman, P., Kumar, K., and Bello, J. P. Wav2clip: Learning robust audio representations from clip. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4563–4567. IEEE, 2022.

Wu, Y., Chen, K., Zhang, T., Hui, Y., Berg-Kirkpatrick, T., and Dubnov, S. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Yee-King, M. J., Fedden, L., and d'Inverno, M. Automatic programming of vst sound synthesizers using deep networks and other techniques. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(2):150–159, 2018.

Young, H., Du, M., and Bastani, O. Neurosymbolic deep generative models for sequence data with relational constraints. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 37254–37266. Curran Associates, Inc., 2022.

A. Supplementary Analyses

A.1. Generation Time

Iter/Popsize	25	50	100
50	5.49 ± 0.154	9.62 ± 0.452	18.43 ± 0.752
100	10.01 ± 0.194	18.05 ± 0.605	33.40 ± 0.331
300	27.61 ± 0.703	49.94 ± 0.424	97.23 ± 0.469

Table 4. Time (in seconds) for different population sizes (columns) and iteration counts (rows).

In Table 4 we illustrate the optimization times, in seconds, for different numbers of iterations (rows) and optimizer population sizes (columns) below, on a modest GPU, i.e. single V100. Note that the necessary number of iterations varies for different prompts, from 50 to 300+ to get optimal results.

A.2. CLAP Scores

Model	AudioSet-50	ESC-50
AudioGen	0.249 ± 0.160	0.277 ± 0.180
AudioLDM	0.166 ± 0.128	0.173 ± 0.142
CTAG	$\mathbf{0.573} \pm 0.126$	$\mathbf{0.585} \pm 0.130$
Real	–	0.416 ± 0.139

Table 5. Comparison of CLAP scores between *CTAG* and other generative models on AudioSet-50 and ESC-50 datasets

Table 5 shows the CLAP (Wu et al., 2023) evaluations for each model with AudioSet-50 and ESC-50 prompts, as well as for the actual ESC-50 dataset of real sounds. CLAP is the objective that we optimize in our synthesis-by-optimization approach, and these results show how *CTAG* trivially achieves a higher score compared to all other models and even the real data. This highlights the ability of our optimization strategy to effectively maximize the CLAP score, and also the importance of finding alternative and distinct evaluation metrics as we showed in Section 3.4.

A.3. Prompting Strategies for All Tested Models

Dataset	Metric	Model	Sounds	Template	Caption
AudioSet-50	Top-1	*AudioGen*	51.6	**57.0**	48.8
		AudioLDM	17.4	**21.0**	16.6
		CTAG	**26.2**	25.2	23.6
	Top-5	*AudioGen*	77.4	**84.8**	80.8
		AudioLDM	44.2	**49.8**	48.0
		CTAG	45.2	**52.2**	51.6
ESC-50	Top-1	*AudioGen*	54.0	**69.0**	62.0
		AudioLDM	23.0	20.2	**29.4**
		CTAG	**16.4**	11.4	13.8
	Top-5	*AudioGen*	71.8	**85.2**	81.8
		AudioLDM	49.4	47.0	**58.4**
		CTAG	30.2	26.4	**31.0**

Table 6. Performance comparison, with different prompting strategies, of models on AudioSet-50 and ESC-50 datasets

For completeness, Table 6 provides all the results for all different models with templates and captions as we showed for *CTAG* in Section 4.3. The performance of *AudioGen* shows a notable boost when using the +T (Template) strategy. However, the impact of these strategies on the other models and datasets is less consistent, with some cases showing modest

improvements and others exhibiting a decrease in performance (e.g., *AudioLDM* ESC-50 +T, *AudioLDM* AudioSet-50 +C). Given the variability in results, it is difficult to make a definitive statement about the effectiveness of these strategies across all baselines. While they may prove beneficial in certain scenarios, their impact appears to be context-dependent.

A.4. User Study Statistical Models

We report post-hoc contrasts for the user study results in Tables 7 to 9.

contrast	odds.ratio	SE	asymp.LCL	asymp.UCL	z.ratio	p.value
AudioLDM / AudioGen	0.31	0.07	0.19	0.53	-5.28	<1e-04
CTAG / AudioGen	0.85	0.18	0.51	1.42	-0.75	1
CTAG / AudioLDM	2.72	0.59	1.61	4.58	4.59	<1e-04

Table 7. Post-hoc contrasts from a mixed-effects logistic regression for accuracy.

contrast	estimate	SE	df	lower.CL	upper.CL	t.ratio	p.value
AudioLDM - AudioGen	-0.53	0.12	579	-0.82	-0.24	-4.34	<1e-04
CTAG - AudioGen	-0.48	0.12	579	-0.78	-0.19	-3.97	0.00024
CTAG - AudioLDM	0.04	0.12	579	-0.25	0.34	0.37	1

Table 8. Post-hoc contrasts from a mixed-effects linear regression for confidence ratings.

contrast	estimate	SE	df	lower.CL	upper.CL	t.ratio	p.value
AudioLDM - AudioGen	0.57	0.12	579	0.29	0.86	4.81	<1e-04
CTAG - AudioGen	1.22	0.12	579	0.93	1.51	10.20	<1e-04
CTAG - AudioLDM	0.65	0.12	579	0.36	0.93	5.39	<1e-04

Table 9. Post-hoc contrasts from a mixed-effects linear regression for artistic interpretativeness.

A.5. User Study Per-Prompt Accuracy

Figure 5 shows the accuracy of our user study participants at classifying sounds generated with *CTAG*, *AudioGen*, and *AudioLDM*. Reviewing these differences shows that some sounds are overall more difficult to identify, for instance; "Truck air brake". This may be due to the ambiguity in what this can sound like, as it is not as common a sound as "Bicycle bell".

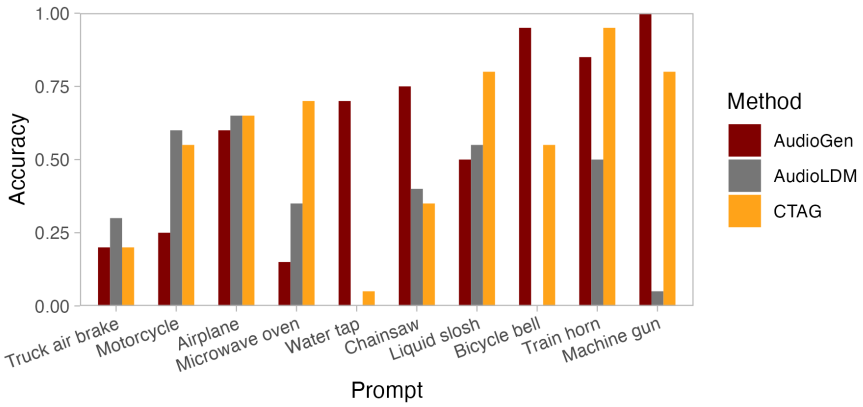

Figure 5. User study classification accuracy per prompt, for *CTAG*, *AudioGen*, and *AudioLDM*.

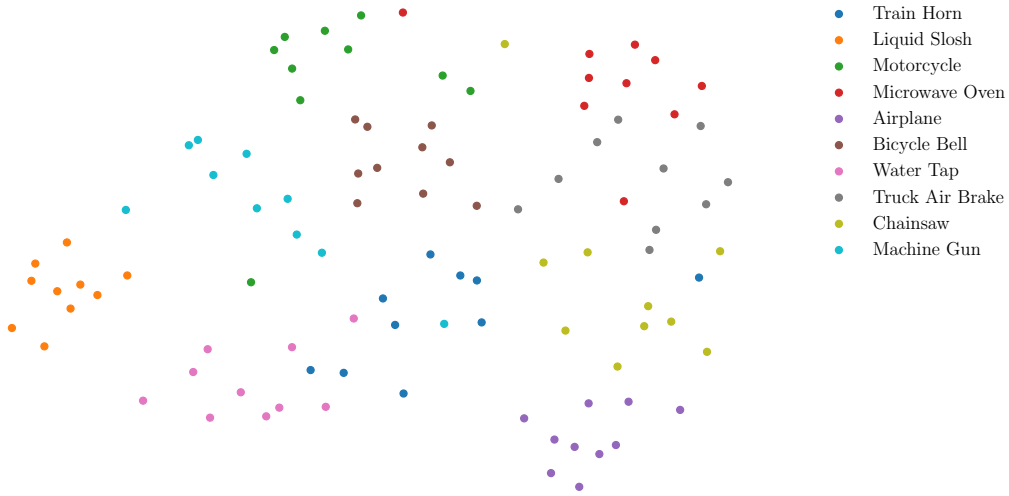

Figure 6. Dimensionality reduction of the *Voice* synthesizer parameters using UMAP applied to 10 sounds from each of the 10 classes from the user study. I t distinctly reveals clusters corresponding to individual sounds, and it shows how conceptually similar sounds such as "water tap" and "liquid slosh" are closer in space.

A.6. Dimensionality Reduction

Having access to the parameters of the synthesizer also allows us to project them into a two-dimensional space to explore the relationship between sounds. Leveraging the Uniform Manifold Approximation and Projection (UMAP) (McInnes et al., 2018) algorithm for dimensionality reduction of the synthesizer parameters, Figure 6 shows how the representation delineates clusters for each distinct sound class while retaining semantic meaning—sounds with similar acoustic properties cluster together.

B. Caption Prompt

We used the following instructions to generate caption-like prompts from class labels:

> *"Write a simple one-sentence audio caption that describes objectively each sound itself in a real scenario without making up any extra details about other possible sounds or places. You should define the most common action for such an entity when multiple options are available. Avoid using templates such as 'A sound of' or 'The sound of'. Sounds: [List]"*

C. Listener Survey

In this section, we provide information about the survey design we used to collect human ratings.

C.1. Survey Flow

- Standard: Introduction (3 Questions)
- Block: Audio (4 Questions)
- Standard: Additional (2 Questions)

C.2. Start of Block: Introduction

Q1: We are conducting a survey to assess the quality of a novel method for text-to-audio generation. You will be presented with a series of short sounds, and asked to select the closest category from a given list, the confidence in your prediction, and how artistically designed the sound is compared to a more realistic interpretation.

Q2: I consent to participate. I understand that my participation is voluntary and I may withdraw my consent at any time.

- Yes (1)
- No (2)

Q3: I am at least 18 years old.

- Yes (1)
- No (2)

Q4: Do you have any hearing loss or hearing difficulties?

- Yes (1)
- No (2)

Q5: Are you fluent in English?

- Yes (1)
- No (2)

Q5: What is your Prolific ID? Please note that this response should auto-fill with the correct ID

C.3. Start of Block: Audio

We use Qualtrics' Loop & Merge functionality to loop through the sounds.

A: Select the closest category for the following sound: **[Audio Clip]**

- Truck air brake (1)
- Water tap (2)
- Train horn (3)
- Motorcycle (4)
- Microwave oven (5)
- Liquid slosh (6)
- Chainsaw (7)
- Airplane (8)
- Bicycle bell (9)
- Machine gun (10)

B: How confident are you in your selected answer?

- Completely confident (1)
- Fairly confident (2)
- Somewhat confident (3)

- Slightly confident (4)
- Not confident at all (5)

C: Would you associate this sound more with a realistic portrayal or an artistic interpretation of the category that you selected?

- 1 (1) Realistic Portrayal
- 2 (2) •
- 3 (3) •
- 4 (4) •
- 5 (5) Artistic Interpretation

C.4. Start of Block: Additional

We have two questions to check that participants were paying attention.

A1 Please select ”Chainsaw” from the options below:

- Truck air brake (1)
- Water tap (2)
- Train horn (3)
- Motorcycle (4)
- Microwave oven (5)
- Liquid slosh (6)
- Chainsaw (7)
- Airplane (8)
- Bicycle bell (9)
- Machine gun (10)

A2: All of the sounds you heard during this survey were the same.

- Yes (1)
- No (2)

Completion Message: Thank you for taking part in this study. Please click the button below to be redirected back to Prolific and register your submission.