AWOL for audio: Text-to-Audio Synthesis via Conditional Real-NVP

Giuseppe Bellantone 1883846

Abstract

AWOL (Analysis WithOut synthesis using Language) is a method for generating 3D shapes (rigged animals and procedural trees) by learning a mapping from CLIP latent space to parametric model parameters. In this project, I try to do the same for audio, learning a mapping from CLAP latent space to synthetizer parameters.

1. Introduction

I present a minimal, end-to-end pipeline that converts short textual prompts (e.g., "a guitar note") into parametric audio and finally a .wav file. The idea is to adapt the AWOL-style latent-space mapping to the audio domain: we learn a mapping between CLAP text embeddings and a compact synthesis parameter vector (pitch, velocity, ADSR envelope, and simple FM controls). The mapping is implemented with a conditional Real-NVP flow trained with a reconstruction + cycle consistency objective. The model stabilizes and enables prompt-driven audio parameter generation that can be rendered directly to sound.

2. Related works

Text-conditioned 3D generation has shown that language embeddings can control structured parametric spaces (like in AWOL). In audio, CLAP provides a joint text–audio embedding space; text-to-audio synthesis has been explored by diffusion or autoregressive models (e.g., AudioLDM/MusicLM), which generate raw audio but do not expose interpretable parameters. This work targets a complementary point: language-to-parameters, yielding interpretable controls and fast, deterministic rendering.

3. Method

3.1. Data and Targets

We build a paired dataset from the NSynth-test dataset (fraction of the larger Nsynth dataset). Nsynth is a huge dataset of audio files each of the same length (around 4 seconds), each one contains a single note of an instrument and each file is paired with data and metadata (pitch_midi, velocity and sample rate were already given in the dataset). Starting from NSynth-test, I deleted all the sythetized instruments, keeping only the real instruments notes for this project. Since the instrument with the less samples was samples (the flute) only had 55 samples, to balance the training I randomly chose 55 samples from the other instruments, remaining with 495 samples (9 instruments in total). The paired dataset is formed by:

Captions: prompts like "a guitar note" (just like in AWOL).
Audio: associated `.wav`.
Targets (θ): a 10-dim vector:

$$\begin{aligned}\theta = [&\text{pitch\_midi}, \text{velocity}, \text{sample\_rate}, \text{duration}_s,\\ &\text{attack}_s, \text{decay}_s, \text{sustain}, \text{release}_s,\\ &\text{mod\_ratio}, \text{mod\_index}] \qquad (1)\end{aligned}$$

ADSR is estimated from the analytic envelope (Hilbert + smoothing) and FM ratio/index from spectral structure around the carrier.

3.2. Model: Conditional Real-NVP

We model a bijection $f_\phi : \theta \leftrightarrow z$ conditioned on text c with affine coupling layers and learned masks. Each coupling layer predicts scale/shift with a residual MLP (ReLU, LayerNorm, dropout) on $[m \odot \theta, c]$, permuting dimensions across layers.

Forward / inverse

$$z = f_\phi(\theta \mid c)$$

Email: Giuseppe Bellantone <bellantone.1883846@studenti.uniroma1.it>.

Deep Learning and Applied AI 2025, Sapienza University of Rome, 2nd semester a.y. 2024/2025.

$$\hat{\theta} = g_\phi(z \mid c) = f_\phi^{-1}(z \mid c)$$

Training losses
Let θ_n, c_n denote normalized targets and conditions. We optimize the following objective:

$$\mathcal{L} = \underbrace{\left\| g_\phi(0 \mid c_n) - \theta_n \right\|_{\text{Huber}}}_{\text{reconstruction}} + \lambda_{\text{cyc}} \underbrace{\left\| g_\phi(f_\phi(\theta_n \mid c_n) \mid c_n) - \theta_n \right\|_{\text{Huber}}}_{\text{cycle}} + \lambda_z \underbrace{\left\| f_\phi(\theta_n \mid c_n) \right\|_2^2}_{\text{Gaussian prior}}. \quad (2)$$

We use AdamW as optimizer, EMA (exponential moving average) of weights, small Gaussian noise on c_n for robustness, and a ReduceLROnPlateau scheduler on the validation objective.

3.3. Contribution

The baseline for the model was the AWOL structure.
The final system combines:
- Cycle consistency,
- Latent prior,
- EMA + text noise,
- End-to-end inference: Caption $\rightarrow$ CLAP $\rightarrow$ Flow $\rightarrow$ θ $\rightarrow$ Synthesizer $\rightarrow$ `.wav`.

4. Results and Experiments

The dataset was split in 90/10 train/val (random with fixed seed).

I followed the same experimental setup as AWOL, training a conditional Real-NVP with the same loss structure. The model is trained with a reconstruction (L1) loss, avoiding the canonical density loss used in normalizing flows. Training is run until the loss stabilizes.

Training was run for 3500 epochs, which corresponds to the point where the validation loss stabilized (similar to the 6000-epoch convergence reported in AWOL). Beyond this point, no further improvements were observed.

In terms of qualitative results, the synthesized audio obtained from textual captions is not yet high-fidelity, due to the simplicity of the FM synthesizer used as a proxy. However, the model was able to capture meaningful correspondences between text embeddings and synthesis parameters. For example, interpolations such as *”a note of a mandolin”* and *”a note of a guitar”* produced perceptually similar sounds, indicating that the model learned a semantically structured latent space.

Testing with the parameters of the net, I got the best results 10 coupling layers and with a batch size of 128.

4.1. Limitations

- Caption granularity is intentionally minimal to match the ones in AWOL; this can limit precision on fine-grained parameters (e.g., exact ADSR times).
- Target noise: FM ratio/index are rough spectral heuristics—errors in labels constrain the upper bound of performance.

5. Conclusions

We have addressed the problem of generating audio from natural language captions using parametric synthesis models. Following AWOL, we used minimun captions and tried to map them in the space of audio parameters. We did this by applying a conditional Real-NVP flow to map CLAP text embeddings to a compact set of synthesis parameters (ADSR + FM). Our qualitative experiments confirm that, while the generated audio is not always perfect, the system achieves meaningful interpolation across instrument families (e.g., “a note of a mandolin” sounding perceptually close to “a note of a guitar”). Future works could focus on trying more complex and specific captions, with more information about the sound (”a high pitched note of a guitar”) or maybe try to generate real world sounds with smilar, models.

A. Appendix

Before I started working with nsynth and instruments, I was trying to work with real world sounds. I was using the ESC-50 dataset, which contained lot of different category of sounds, all with the same length. After a lot of failed models, I realized that those sounds (”a dog barking”, a baby’s crying”, etc.) were actually to difficult to replicate with a simple synthetizer, and it was hard to imagine to be able to make interpolation on them, so I switched to Nsynth, since in it audios are just a single instrument note which is easier to replicate and work with.

References

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 *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2022.

Engel, J., Resnick, C., Roberts, A., Dieleman, S., Eck, D., Simonyan, K., and Norouzi, M. Nsynth: A neural audio synthesis dataset. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2017. URL https://magenta.tensorflow.org/datasets/nsynth.

Liu, H., Chen, Q., and et al. Audioldm: Text-to-audio generation with latent diffusion models. *Proc. of the International Conference on Machine Learning (ICML)*, 2023. URL https://arxiv.org/abs/2301.12503.

Silvia Zuffi, M. J. B. Awol: Analysis without synthesis using language. 2023. URL https://doi.org/10.48550/arXiv.2404.03042.

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 *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2023.

(Silvia Zuffi, 2023) (Engel et al., 2017) (Wu* et al., 2023) (Chen et al., 2022) (Liu et al., 2023)