Incorporating Music Knowledge in Continual Dataset Augmentation for Music Generation

Alisa Liu¹, Alexander Fang¹, Gaëtan Hadjeres², Prem Seetharaman¹, Bryan Pardo¹
1. Northwestern University, Evanston, IL 2. Sony CSL, Paris, France

Paper: interactiveaudiolab.github.io/assets/papers/Fang2020-MLMD.pdf
Code: github.com/asdfang/constraint-transformer-bach

MOTIVATION
Deep models for music generation require a large training set, which are often lacking for specific musical domains.

BIG TAKEAWAY
Can a generative system be improved by training on its own output?
• Key intuition: training data for a generative system can be augmented by examples it produces during training, provided they are of sufficiently high quality and variety
• We develop the first method to continuously augment training set with generated output
• Our method increases the quality of generated examples on the task of producing Bach-style four-part chorales

AUGMENTATIVE GENERATION

Augmentative Generation (Aug-Gen): a method of dataset augmentation for any music generation system trained on a resource-constrained domain

Aug-Gen Algorithm:
Train a model on \( m \) batches of size \( k \) from training set \( T \)
Generate \( N \) examples with the model
Select a set of high-quality, diverse generated examples \( G \)
Augment \( T \) with \( G \)
Repeat until validation loss minimizes

EXPERIMENTS
Evaluate the effectiveness of Aug-Gen in improving the output quality of a Transformer model trained to generate Bach-style chorales

Experimental Setup
• Generative model: a transformer network with relative attention
• Grading function from (Fang et al., 2020)
• Simple uniqueness criterion for diversity
• In generation step of each epoch, generate \( N = 50 \) chorales
• In training step, train on \( m = 2048 \) randomly selected batches of size \( k = 8 \)
• Train for 40 epochs, and use epoch with lowest validation loss as final model

Compare three training methods that differ only in the threshold \( t \) for including generated chorales in the training set:

- Aug-Gen
- \( t = 0 \) of Bach grades: includes only generated chorales that receive a better grade than 25% of Bach chorales
- \( t = -\infty \): includes no generated chorales, equivalent to training a model on only Bach chorales
- \( t = \infty \): includes all generated chorales, regardless of quality

Aug-Gen results in better generative output, as seen by its tighter grade distribution that more closely resembles Bach’s

FUTURE WORK
• Improve the grading function to account for remaining limitations in generated music
• Explore richer measures of diversity within a musical dataset
• Apply Aug-Gen to different models and musical domains
• Devise other training methods that utilize generated music data

REFERENCES