Incorporating Music Knowledge in Continual Dataset Augmentation for Music Generation

Alisa Liu[1], Alexander Fang[1], Gaëtan Hadjeres[2], Prem Seetharaman[1], Bryan Pardo[1]

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 $\mathbf{T}$

Generate N examples with the model

Select a set of high-quality, diverse generated examples $\mathbf{G}$

Augment $\mathbf{T}$ with $\mathbf{G}$

Repeat until validation loss minimizes

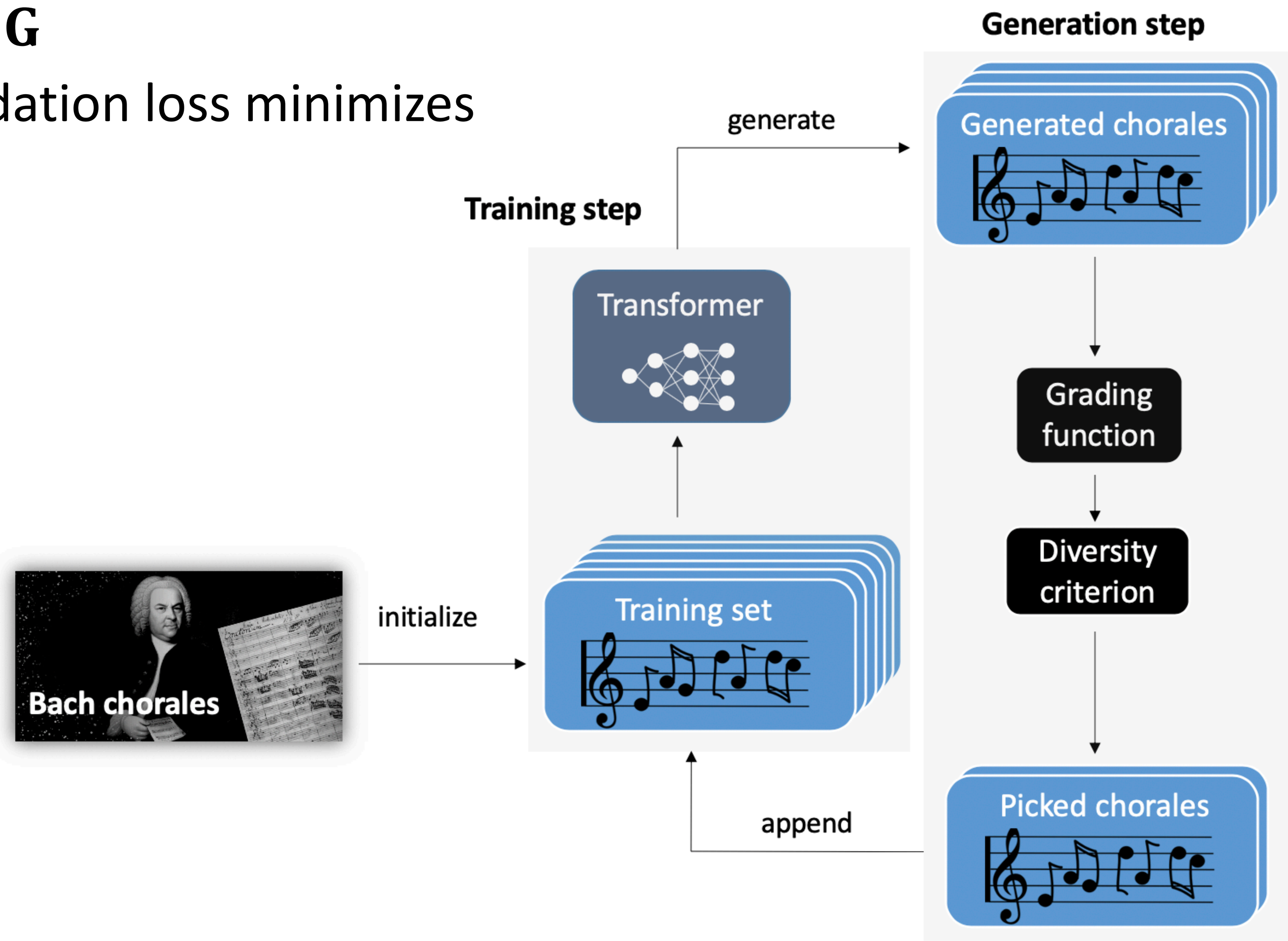

Figure 1: Diagram of the augmentative generation method

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

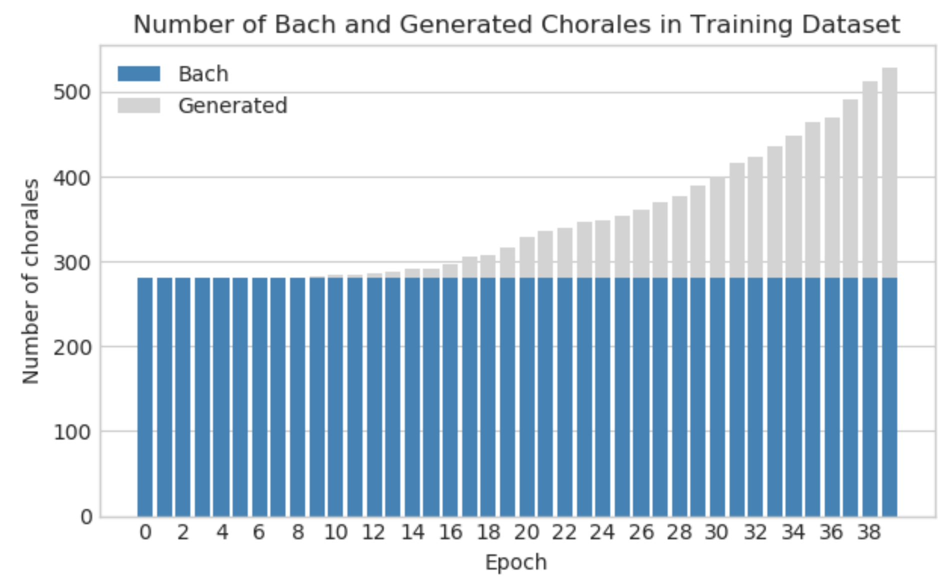

Figure 2 (top-right): The cumulative number of Bach chorales and generated chorales in the dataset during Aug-Gen training

Aug-Gen allows for longer training

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

Aug-Gen	
$t = Q_3$ of Bach grades	Includes only generated chorales that receive a better grade than 25% of Bach chorales
Baseline-none	
$t = -\infty$	Include no generated chorales, equivalent to training a model on only Bach chorales
Baseline-all	
$t = \infty$	Includes all generated chorales, regardless of quality

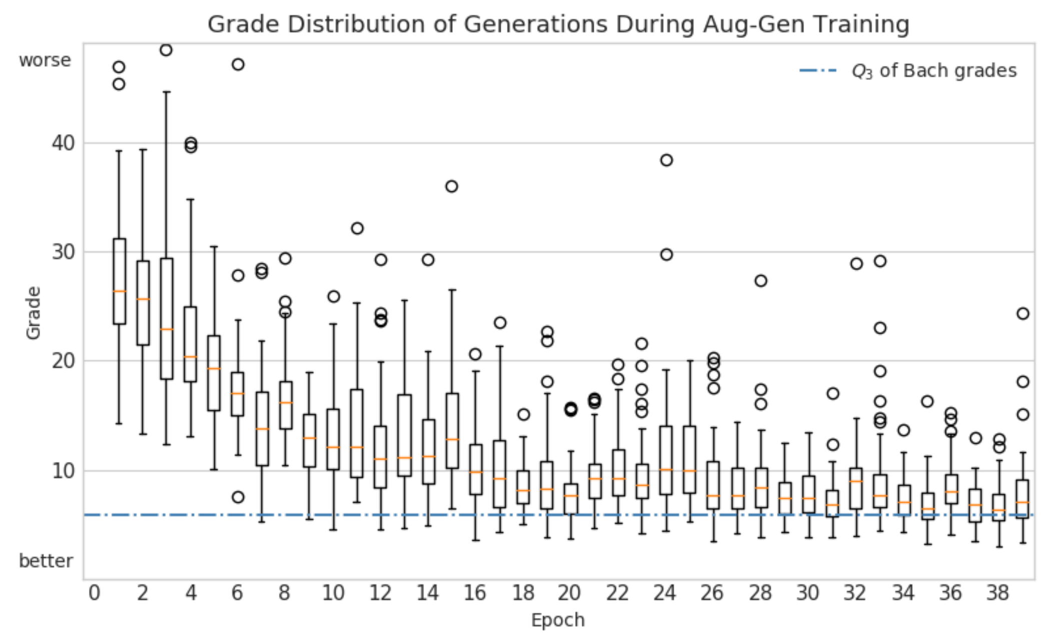

Figure 3 (right): A series of boxplots representing the grade distribution of 50 chorales generated at each epoch of training in Aug-Gen

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

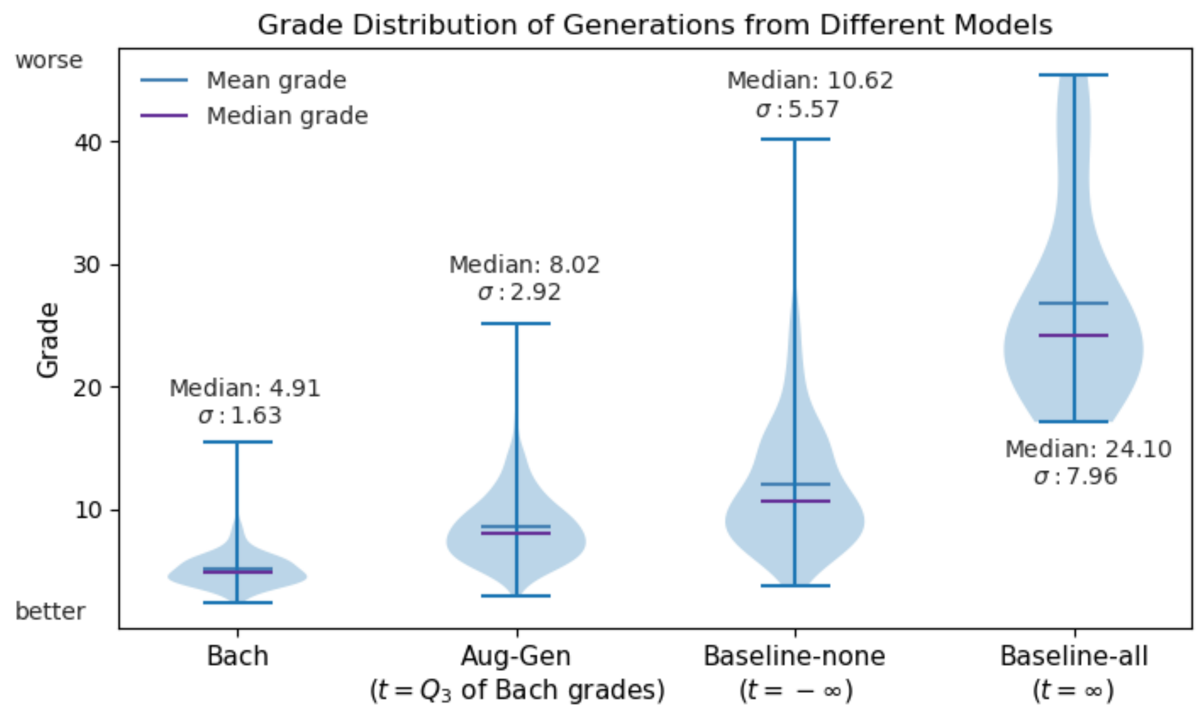

Figure 4 (bottom-left): The grade distribution of the 351 Bach chorales, and 351 output generated from each model

Remaining errors tend to be along dimensions not measured by the grading function, e.g. excessive modulation, weak metric structure, unmusical repetition

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

Alexander Fang, Alisa Liu, Prem Seetharaman, Bryan Pardo. Bach or Mock? A Grading Function for Chorales in the Style of J.S. Bach. In *The Machine Learning for Media Discovery (ML4MD) Workshop* in conjunction with ICML 2020.