# COUNTERPOINT BY CONVOLUTION

**Cheng-Zhi Anna Huang**[*] **& Tim Cooijmans**[†]
Department of Computer Science and Operations Research
University of Montreal
Montreal, MC H3T 1N8, Canada
{chengzhiannahuang,cooijmans.tim}@gmail.com

**Adam Roberts**
Google Brain
Mountain View, CA 94043, U.S.
adarob@google.com

**Aaron Courville**
Department of Computer Science & Operations Research
University of Montreal
Montreal, MC H3T 1N8, Canada
aaron.courville@umontreal.ca

**Douglas Eck**
Google Brain
Mountain View, CA 94043, U.S.
deck@google.com

## ABSTRACT

Machine learning models of music typically break down the task of composition into a chronological process, composing a piece of music in a single pass from beginning to end. On the contrary, human composers write music in a nonlinear fashion, scribbling motifs here and there, often revisiting choices previously made. We reformulate musical composition as an inpainting task and introduce COCONET, a convolutional neural network in the NADE family of generative models. However, the NADE ancestral sampling procedure produces poor samples, and we explore two alternative sampling procedures based on blocked Gibbs sampling. We demonstrate the versatility of our method on three generative tasks: conditioned rewriting, partial score completion, and unconditioned polyphonic music generation. Performance is evaluated based on likelihood estimates and user studies.

## 1 INTRODUCTION

Machine learning can be used to create compelling art. This was shown recently by DeepDream (Mordvintsev et al., 2015), an optimization process that created psychedelic transformations of images. A similar idea underlies a variety of style transfer algorithms (Gatys et al., 2015), which impose textures and colors from one image onto another. More recently, the multistyle pastiche generator (Dumoulin et al., 2016) exposes adjustable knobs that allow users of the system fine-grained control over style transfers. Neural doodle (Champandard, 2016) further closes the feedback loop between algorithm and artist.

We wish to bring similar artistic tools to the domain of music. Whereas previous work in music has relied mainly on sequence models such as Hidden Markov Models (HMMs, Baum & Petrie (1966)) and Recurrent Neural Networks (RNNs, Rumelhart et al. (1988)), we instead employ convolutional neural networks due to their emphasis on capturing local structure and their invariance properties. Moreover, convolutional neural networks have shown to be extremely versatile once trained, as shown by a variety of creative uses in the literature (Mordvintsev et al., 2015; Gatys et al., 2015; Almahairi et al., 2015; Lamb et al., 2016).

[*]Work done while the author was at Google.
[†]Work done while the author was at Google.

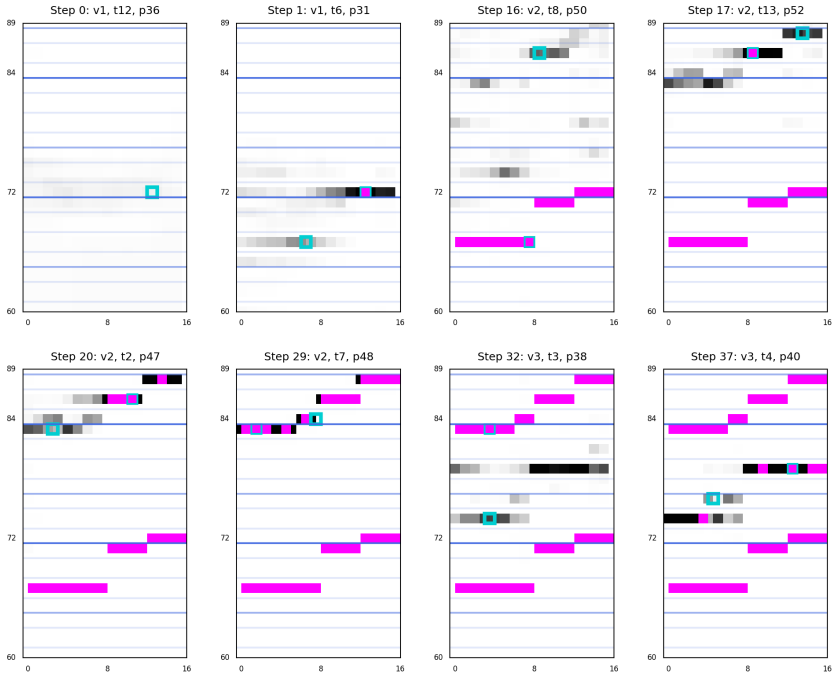


Figure 1: Highlights of pianoroll predictions of various steps in an unconditioned generation of a musical fragment by Coconet

We formulate musical composition as an inpainting task and introduce COCONET, a deep convolutional model trained to reconstruct partial scores. Once trained, COCONET provides direct access to all conditionals of the form $p(\mathbf{x}_{\neg C} \mid \mathbf{x}_C)$ where $\mathbf{x}_C$ is a fragment of a musical score $\mathbf{x}$ and $\mathbf{x}_{\neg C}$ is its complement.

We show that this inpainting setup is closely related to deep orderless NADE Uria et al. (2014), which learns an ensemble of factorizations of the joint $p(\mathbf{x})$. However, the sampling procedure for orderless NADE is not orderless. Sampling from an orderless NADE involves (randomly) choosing an ordering, and sampling ancestrally according to the chosen ordering. We have found that this produces poor results for the highly structured and complex domain of musical counterpoint.

Instead, we propose a novel sampling procedure that wraps ancestral sampling in blocked Gibbs, essentially improving sample quality through rewriting. This approach is related to that of Yao et al. (2014) where a NADE is employed in the transition operator for a Markov Chain, which yields a Generative Stochastic Network (GSN). The transition consists of a corruption process that masks out a subset $\mathbf{x}_{\neg C}$ of variables, followed by a process that independently resamples each variable $\mathbf{x}_i, i \notin C$ according to the distribution $p_\theta(\mathbf{x}_i \mid \mathbf{x}_C)$ emitted by the NADE. The effects of independent sampling are amortized by annealing the probability with which variables are masked out. In our method, on the other hand, we always resample from the joint $p_\theta(\mathbf{x}_{\neg C} \mid \mathbf{x}_C)$ through ancestral sampling. This is much more expensive, but yields much better samples.

We show the versatility of our method through examples and human evaluations on three generative tasks: conditioned rewriting, partial score completion, and unconditioned polyphonic music generation.

Section 2 discusses previous work in the area of automatic musical composition. Section 3 introduces the musical equivalent of image inpainting, the task we train our model to solve. The details of our convolutional model are laid out in Section 4. In Section 5 we show that our approach is

equivalent to that of deep and orderless NADE Uria et al. (2014). We discuss sampling from our model in Section 6. Results of quantitative and qualitative evaluations are reported in Section 7. Finally, Section 8 concludes.

## 2 RELATED WORK

Sequence models such as HMMs and RNNs are a natural choice for modeling music. However, one of the challenges in adapting such models to music is that music generally consists of multiple interdependent streams of events. This can be most clearly seen in the notion of counterpoint, which refers to the relationships between the movement of individual instruments in a musical work. Compare this to typical sequence domains such as speech and language, which involve modeling a single stream of events: a single speaker or a single stream of words.

Successful application of sequence models to music hence requires serializing or otherwise re-representing the music to fit the sequence paradigm. For instance, Liang (2016) serialize four-part Bach chorales by interleaving the parts, while Allan & Williams (2005) construct a chord vocabulary. Boulanger-Lewandowski (2014) adopt a piano roll representation, which is a binary matrix $\mathbf{X}$ such that $x_{it}$ is hot if some instrument is playing pitch $i$ at time $t$. To model the joint probability distribution of the multi-hot pitch vector $\mathbf{x}_t$, they employ a Restricted Boltzmann Machine (RBM (Smolensky, 1986; Hinton et al., 2006)) or Neural Autoregressive Distribution Estimator (Uria et al., 2016) at each time step.

Moreover, the behavior of human composers does not fit the chronological mold assumed by previous authors. A human composer might start his work with a coarse chord progression and iteratively refine it, revisiting choices previously made. Sampling according to $x_t \sim p(x_t|x_{<t})$, as is common, cannot account for the kinds of timeless dependencies that composers employ. Hadjeres et al. (2016) sidestep the choice of causal factorization and instead employ an undirected Markov model to learn pairwise relationships between neighboring notes up to a specified number of steps away in a score. Sampling involves Markov Chain Monte Carlo (MCMC) using the model as a Metropolis-Hastings (MH) objective. The model permits constraints on the state space to support tasks such as melody harmonization. However, the Markov assumption severely limits the expressivity of the model.

We opt instead for a convolutional approach that avoids many of these issues and naturally captures both relationships across time and interactions between instruments.

## 3 MUSICAL COMPOSITION AS INPAINTING

We consider the musical equivalent of inpainting, a versatile setting that generalizes popular tasks such as melody harmonization, partial score completion and composition from scratch. Inpainting (Bertalmio et al., 2000) is the task of restoring damaged or missing parts of an image. In machine learning, image inpainting has found popularity as an unsupervised learning task, where a model is trained to reconstruct an image after it has been corrupted by a random process (Pathak et al., 2016).

Inpainting readily carries over to music when we view it as a stack of piano rolls represented by the binary three-tensor $\mathbf{x} \in \{0, 1\}^{I \times T \times P}$. Here $I$ denotes the number of instruments, $T$ the number of time steps, $P$ the number of pitches, and $\mathbf{x}_{i,t,p} = 1$ iff the $i$th instrument plays pitch $p$ at time $t$. We will assume each instrument plays exactly one pitch at a time, that is, $\sum_p \mathbf{x}_{i,t,p} = 1$ for all $i, t$.

For the present work we will restrict ourselves to the study of four-part Bach chorales as used in prior work (Allan & Williams, 2005; Boulanger-Lewandowski, 2014; Goel et al., 2014; Liang, 2016; Hadjeres et al., 2016). Hence we assume $I = 4$ throughout. We discretize pitch according to equal temperament, but constrain ourselves to only the range that appears in our training data (MIDI pitches 36 through 88). Time is discretized at the level of 16th notes for similar reasons.

Given a training example $\mathbf{x} \sim p(\mathbf{x})$, we present the model with the values of only a strict subset of its elements $\mathbf{x}_C = \{\mathbf{x}_{(i,t)} \mid (i, t) \in C\}$ and ask it to reconstruct its complement $\mathbf{x}_{\neg C}$. The loss function is given by

$$\mathcal{L}(\mathbf{x}; C, \theta) = -\sum_{(i,t)\notin C} \log p_\theta(\mathbf{x}_{i,t} \mid \mathbf{x}_C, C) \tag{1}$$

$$= -\sum_{(i,t)\notin C} \sum_p \mathbf{x}_{i,t,p} \log p_\theta(\mathbf{x}_{i,t,p} \mid \mathbf{x}_C, C) \tag{2}$$

where $p_\theta$ refers to the probability under the model parameterized by $\theta$. We wish to minimize the expected loss

$$\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\mathbb{E}_{C\sim p(C)}\mathcal{L}(\mathbf{x}; C, \theta). \tag{3}$$

## 4 COUNTERPOINT BY CONVOLUTION

We approach the task outlined above using a deep convolutional neural network (Krizhevsky et al., 2012). This choice is motivated by the locality of contrapuntal rules and their near-invariance to translation, both in time and in the frequency spectrum.

The input to the model is obtained by masking the piano rolls $\mathbf{x}$ to obtain the context $\mathbf{x}_C$ and concatenating this with the corresponding mask:

$$\mathbf{h}^0_{i,t,p} = \mathbb{1}_{(i,t)\in C}\mathbf{x}_{i,t,p} \tag{4}$$

$$\mathbf{h}^0_{I+i,t,p} = \mathbb{1}_{(i,t)\in C} \tag{5}$$

where the first dimension ranges over channels and the time and pitch dimensions are convolved over.

$$\mathbf{a}^l = \mathrm{BN}(\mathbf{W}^l * \mathbf{h}^{l-1}; \gamma^l, \beta^l) \tag{6}$$

$$\mathbf{h}^l = \mathrm{ReLU}(\mathbf{a}^l + \mathbf{h}^{l-2}) \qquad \text{for } 3 < l < L-1 \text{ and } l = 0 \bmod 2 \tag{7}$$

$$\mathbf{h}^L = \mathbf{a}^L \tag{8}$$

With the exception of the first and final layers, all of our convolutions preserve the size of the input. That is, we use "same" padding throughout and all activations $h^l, 1 < l < L$ have 128 channels. The network consists of 64 layers with $3 \times 3$ filters on each layer. After each convolution we apply batch normalization Ioffe & Szegedy (2015) (denoted by $\mathrm{BN}(\cdot)$) with statistics tied across time and pitch. After every second convolution, we introduce a skip connection from the hidden state two levels below to reap the benefits of residual learning He et al. (2015).

Finally, we obtain predictions for the pitch at each instrument/time pair:

$$\hat{p}_\theta(\mathbf{x}_{i,t,p} \mid \mathbf{x}_C, C) = \frac{\exp(h^L_{i,t,p})}{\sum_p \exp(h^L_{i,t,p})} \tag{9}$$

Based on the predictions, we compute the loss (Equation 1) and optimize it with respect to the parameters $\theta = \mathbf{W}^1, \gamma^1, \beta^1, \ldots, \mathbf{W}^{L-1}, \gamma^{L-1}, \beta^{L-1}$ by stochastic gradient descent with step size determined by Adam (Kingma & Ba, 2014).

## 5 EQUIVALENCE TO ORDERLESS NADE

Our inpainting approach is equivalent to an *orderless and deep* Neural Autoregressive Distribution Estimator (NADE, Uria et al. (2016)). NADE models a $d$-variate distribution $p(\mathbf{x})$ through a factorization

$$p_\theta(\mathbf{x}) = \prod_d p_\theta(\mathbf{x}_{o_d} \mid \mathbf{x}_{o_{<d}}) \tag{10}$$

where $o$ is a permutation, and the parameters $\theta$ are shared among the conditionals. NADE can be trained for all orderings $o$ simultaneously using the orderless NADE (Uria et al., 2014) training procedure. This procedure relies on the observation that, thanks to parameter sharing, computing $p_\theta(\mathbf{x}_{o_{d'}} \mid \mathbf{x}_{o_{<d}})$ for all $d' \geq d$ is no more expensive than computing it only for $d' = d$. Hence for a given $o$ and $d$ we can simultaneously obtain partial losses for all orderings that agree with $o$ up to $d$:

$$\mathcal{L}_{\text{NADE}}(\mathbf{x}; o_{<d}, \theta) = -\sum_{o_d} \log p_\theta(\mathbf{x}_{o_d} \mid \mathbf{x}_{o_{<d}}, o_{<d}, o_d) \tag{11}$$

$$\tag{12}$$

Letting $o_{<d} = C$, we obtain our loss from Equation 1

$$\mathcal{L}_{\text{COCONET}}(\mathbf{x}; C, \theta) = -\sum_{(i,t) \notin C} \log p_\theta(\mathbf{x}_{i,t} \mid \mathbf{x}_C, C) \tag{13}$$

For any one sample $(\mathbf{x}, C)$, this loss consists of $|\neg C|$ terms of the form $\log p_\theta(\mathbf{x}_{i,t} \mid \mathbf{x}_C, C)$. We let $p(C)$ be uniform in the size of the mask and reweight the sample losses according to

$$\tilde{\mathcal{L}}(\mathbf{x}; C, \theta) = \frac{1}{|\neg C|} \mathcal{L}(\mathbf{x}; C, \theta). \tag{14}$$

This correction, due to Uria et al. (2014), ensures consistent estimation of the negative log-likelihood of the joint $p_\theta(\mathbf{x})$.

For the task of inpainting, we might wish to increase the difficulty by choosing $p(C)$ so as to frequently mask out large contiguous regions, as otherwise the model might learn only superficial local relationships. This is discussed in Pathak et al. (2016) for the case of images, where a model might learn only that pixels are similar to their neighbors. Similar low-level relationships hold in our case, as our piano roll representation is binary and very sparse. For instance, if we mask out only a single sixteenth step in the middle of a long-held note, reconstructing the masked out step does not require any deep understanding of music. To this end we also consider choosing the context $C$ by independent Bernoulli samples, such that each variable has a low probability of being included in the context.

## 6 SAMPLING

We can sample from the model using the NADE ancestral ordering procedure. However, we find that this yields poor samples, and we propose instead to use Gibbs sampling.

### 6.1 NADE SAMPLING

To sample according to NADE, we start with an empty (zero everywhere) piano roll $\mathbf{x}^0$ and context $C^0$ and populate them iteratively by the following process. We feed the piano roll $\mathbf{x}^s$ and context $C^s$ into the model to obtain a set of categorical distributions $p_\theta(\mathbf{x}_{i,t} | \mathbf{x}^s_{C^s}, C^s)$ for $(i,t) \notin C^s$. As the $\mathbf{x}_{i,t}$ are not conditionally independent, we cannot simply sample from these distributions independently. However, if we sample from one of them, we can compute new conditional distributions for the others. Hence we randomly choose one $(i,t)^{s+1}$ to sample from, and let $\mathbf{x}^{s+1}_{i,t}$ equal the one-hot realization. Augment the context with $C^{s+1} = C^s \cup (i,t)$ and repeat until the piano roll is populated. This procedure is easily generalized to tasks such as melody harmonization and partial score completion by starting with a nonempty piano roll.

Unfortunately, samples thus generated are of low quality, which we surmise is due to accumulation of errors. While the model provides conditionals $p_\theta(\mathbf{x}_{i,t}|\mathbf{x}_C, C)$ for all $(i,t) \notin C$, some of these conditionals may be better modeled than others. We suspect in particular those conditionals used early on in the procedure, for which the context $C$ consists of very few variables. Moreover, although the model is trained to be order-agnostic, different orderings invoke different distributions, which is another indication that some conditionals are poorly learned.

### 6.2 Gibbs Sampling

To remedy this, we allow the model to revisit its choices: we repeatedly mask out some part of the piano roll and then repopulate it. This is a form of blocked Gibbs sampling (Liu, 1994), where each block is itself sampled sequentially using ancestral sampling as described above. The complete procedure is specified by Algorithm 1.

**Algorithm 1** Sampling from the model by blocked Gibbs.

Given an initial piano roll $\mathbf{x}$ and context $C$
**loop** $n$ times
  **if** not the first iteration **then**
    Choose $C \sim p(C)$ ▷ Determine a block to resample
    $\mathbf{x}_{i,t} \leftarrow 0 \; \forall (i,t) \notin C$
  **end if**
  **while** $\neg C$ nonempty **do** ▷ Sample from the joint distribution of $\mathbf{x}_{\neg C}$
    Choose $(i,t) \sim \mathrm{Uniform}(\neg C)$
    Choose $p \sim p_\theta(\mathbf{x}_{i,t,p}|\mathbf{x}_C, C)$
    $\mathbf{x}_{i,t,p} \leftarrow 1$
    $C \leftarrow C \cup (i,t)$
  **end while**
**end loop**

Blocked sampling is crucial for mixing, as the high temporal resolution of our representation causes strong correlations between consecutive notes. For instance, without blocked sampling, it would take many steps to snap out of a long-held note. Similar observations hold for the Ising model from statistical mechanics, leading to the development of the Swendsen-Wang algorithm (Swendsen & Wang, 1987) in which large clusters of variables are resampled at once.

Our blocked Gibbs sampling procedure resembles that of Yao et al. (2014), except that they sample the variables within a block independently. To ensure the Gibbs process produces samples from the model distribution $p_\theta(x)$, they anneal the masking probability. Initially, when the masking probability is high, the chain mixes fast but samples are poor due to independent sampling. As the masking probability reduces, fewer variables are sampled at a time, until finally variables are sampled one at a time and conditioned on all the others. We on the other hand always sample variables one by one within the block.

## 7 Evaluation

We evaluate our approach on a corpus of four-part Bach chorales. The literature features many variants of this dataset (Allan & Williams, 2005; Boulanger-Lewandowski, 2014; Liang, 2016; Hadjeres et al., 2016), and we follow the unfortunate tradition of introducing our own adaptation. Although this complicates comparisons against earlier work, we feel justified in doing so as our approach requires instruments to be separated, and other authors' eighth-note temporal resolution is too coarse to accurately convey counterpoint.

We rebuilt our dataset from the Bach chorale musicXML scores readily available through (Cuthbert & Ariza, 2010), which was also the basis for the dataset used in (Liang, 2016). The scores included 357 four-part Bach chorales. We excluded scores that included note durations less than sixteenth notes, resulting in 354 pieces. These pieces were split into train/valid/test in 60/20/20% ratios.

We compare with Liang (2016) based on note-level likelihood and Boulanger-Lewandowski (2014) based on frame-level likelihood. Note that train/valid/test differs among both prior work and also with our work, and that Liang (2016) uses a 80/10/10% split instead.

However, evaluation of generative models is hard (Theis et al., 2015). The gold standard for evaluation is qualitative comparison by humans, and we therefore report results of a human evaluation study. Our model is able to achieve compelling results on three tasks: conditioned rewriting, partial score completion, and unconditioned polyphonic generation.

Conditioned rewriting refers to the case where the initial piano roll $\mathbf{x}^0$ is taken from the validation set, and the piece is iteratively rewritten using Algorithm 1. In partial score completion, we similarly initialize the initial piano roll with an existing piece, but mask out part of it and repopulate it using ancestral sampling. Finally, unconditioned generation starts with an empty piano roll and mask, and populates using Algorithm 1.

### 7.1 Evaluating log-likelihood

To estimate the log-likelihood of a datapoint $\mathbf{x}$, we follow the orderless NADE approach. That is, we uniformly sample a random ordering $(i_1, t_1), (i_2, t_2), \dots (i_{IT}, t_{IT})$, and compute the notewise log-likelihood according to

$$\log \hat{p}_\theta(\mathbf{x}) = \frac{1}{IT} \sum_{d=1}^{IT} \log p_\theta(\mathbf{x}_{i_d,t_d} \mid \mathbf{x}_{C_{d-1}}, C_{d-1}) \tag{15}$$

where $C_d = \bigcup_{c=1}^{d} \{(i_c, t_c)\}$. Note that we randomly crop each datapoint to be $T$ time steps long before processing it, as this facilitates batch processing.

We repeat this procedure $k$ times and average across all point estimates. The numbers for our models in Table 1 were obtained with $k = 5$.

The process for computing the notewise log-likelihood is akin to teacher-forcing, where at each step of the way the model observes the ground truth for all its previous predictions. To compute the framewise log-likelihood, we instead let the model run free within each frame $t$. This results in a more representative measure of the model's quality as it is sensitive to accumulation of error.

Table 1 lists notewise and framewise likelihoods of the validation data under variants of our model, as well as comparable results from other authors. We include four variants of COCONET that differ in the choice of the distribution $p(C)$ over contexts during training. By *importance sampling* we refer to the orderless NADE strategy discussed in Section 5, in which $p(C)$ is uniform over $|C|$ and the sampled losses are reweighted by $1/|\neg C|$. We also evaluate three variants where the contexts are chosen by biased coin flips, that is, $\Pr((i,t) \in C) = \alpha$, for $\alpha \in 0.5, 0.25, 0.1$. The framewise log-likelihood for $\alpha = 0.5$ is listed as $\infty$ as its estimation repeatedly overflowed.

Overall, COCONET seems to underperform in terms of notewise likelihood, yet perform well in terms of framewise likelihood. Estimating the loss by importance sampling appears to work significantly better than determining the context using independent Bernoulli variables, as one might expect. However, the choice of Bernoulli probability $\alpha$ strongly affects the resulting loss, which suggests that some of the conditionals benefit from more training.

### 7.2 Human evaluations

We are carrying out listening tests on Amazon's Mechanical Turk (MTurk) to assess the capacity of our models in generating musical samples. Please note the results below are still preliminary, and we will update this section as soon as new results come in.

A natural question to answer in a listening test is if music generated by our model can be indistinguishable from music composed by Bach. As the musical background of participants on MTurk is quite varied, we opt for another question, "which musical fragment do you prefer?" to assess the quality of the generated fragments.

Table 1: Negative log-likelihood (NLL) on the test set for the Bach corpus. As discussed in the text, our numbers are not directly comparable to those of other authors due to the use of different splits. Results from Boulanger-Lewandowski (2014) were based on an eighth-note temporal resolution (our resolution is sixteenth notes). Please note that our results are preliminary *validation* likelihoods.

| **Model** | **Notewise NLL** | **Framewise NLL** |
|---|---|---|
| Bachbot (Liang, 2016) | 0.477 | – |
| NADE (Boulanger-Lewandowski, 2014) | – | 7.19 |
| RNN-RBM (Boulanger-Lewandowski, 2014) | – | 6.27 |
| RNN-NADE (Boulanger-Lewandowski, 2014) | – | 5.56 |
| COCONET, i.i.d Bernoulli(0.50) | 0.924 | $\infty$ |
| COCONET, i.i.d Bernoulli(0.25) | 0.655 | 4.48 |
| COCONET, i.i.d Bernoulli(0.10) | 0.812 | 4.66 |
| COCONET, importance sampling | 0.569 | 3.73 |

The study design is as follows: we compare between samples generated by our early models and Bach chorales. The model variants include BERNOULLI(0.5) and BALANCED BY SAMPLING which differ in the choice of the distribution $p(C)$ over contexts used during training, and a DENOISING model that was trained to reconstruct piano rolls from their noisy versions. Noise was introduced by a corruption process that randomly perturbs the pitch at locations in the piano roll generated by a Bernoulli(0.5) distribution. For each of our models, we generate four samples from empty piano rolls. For the Bach set, we randomly crop four samples from the chorale validation set. Resulting in four sets of four sounds each. All of the samples are two measures long, lasting twelve seconds.

For each MTurk hit, two random sounds are presented. These sounds are selected by first randomly choosing two sets, and then randomly choosing one sample from each set. Participants are then asked to rate which one of the two samples they prefer on a Likert scale. The study resulted in 192 ratings, where each model was involved in 92 pairwise comparisons. Figure 2 reports the number of times in a pairwise comparison a model/Bach was more preferred.

We performed post-hoc pairwise comparisons using Wilcoxon Signed Rank test. Bach was preferred over our balanced by sampling model. The pairs that did not show a statistically significant difference include BERNOULI(0.5) vs Bach, DENOISING vs Bach, and BERNOULI(0.5) vs DENOISING.

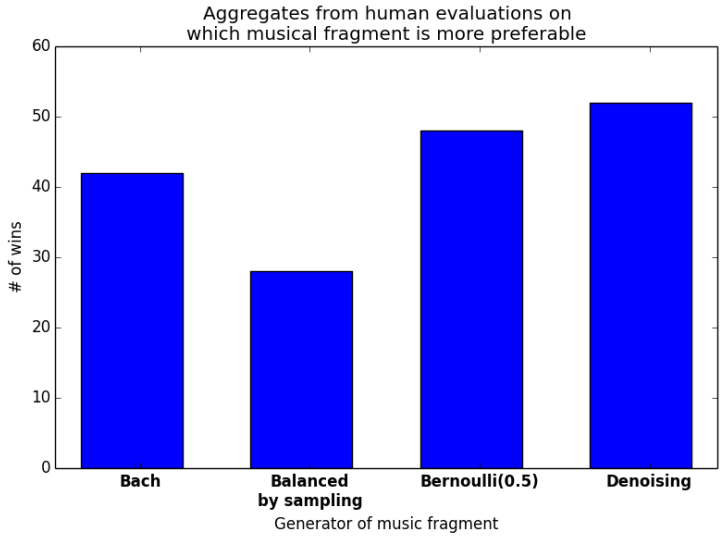


Figure 2: MTurk results from human evaluations on unconditioned generation.

## 8 CONCLUSION

We introduced a convolutional approach to modeling musical scores based on the NADE (Uria et al., 2016) framework and image inpainting Pathak et al. (2016). We've shown that the NADE ancestral sampling procedure yields poor samples for our domain, and argued that this is because some conditionals are not captured well by the model. Our novel Gibbs sampling scheme improves sample quality. Participants in a user study preferred musical fragments generated by our model and those composed by Bach about equally often.

### ACKNOWLEDGMENTS

We thank Kyle Kastner and Guillaume Alain, Curtis (Fjord) Hawthorne, the Google Brain Magenta team, as well as Jason Freidenfelds for helpful feedback, discussions, suggestions and support.

## REFERENCES

Moray Allan and Christopher KI Williams. Harmonising chorales by probabilistic inference. *Advances in neural information processing systems*, 17:25–32, 2005.

Amjad Almahairi, Nicolas Ballas, Tim Cooijmans, Yin Zheng, Hugo Larochelle, and Aaron Courville. Dynamic capacity networks. *arXiv preprint arXiv:1511.07838*, 2015.

Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966.

Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 417–424. ACM Press/Addison-Wesley Publishing Co., 2000.

Nicolas Boulanger-Lewandowski. Modeling high-dimensional audio sequences with recurrent neural networks. 2014.

Alex J. Champandard. Neural doodle, 2016. URL `https://github.com/alexjc/neural-doodle`.

Michael Scott Cuthbert and Christopher Ariza. music21: A toolkit for computer-aided musicology and symbolic music data. 2010.

Vincent Dumoulin, Johnathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

Kratarth Goel, Raunaq Vohra, and JK Sahoo. Polyphonic music generation by modeling temporal dependencies using a rnn-dbn. In *International Conference on Artificial Neural Networks*, pp. 217–224. Springer, 2014.

Gaëtan Hadjeres, Jason Sakellariou, and François Pachet. Style imitation and chord invention in polyphonic music with exponential families. *arXiv preprint arXiv:1609.05152*, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Alex Lamb, Vincent Dumoulin, and Aaron Courville. Discriminative regularization for generative models. *arXiv preprint arXiv:1602.03220*, 2016.

Feynman Liang. Bachbot: Automatic composition in style of bach chorales. *Masters thesis, University of Cambridge*, 2016.

Jun S Liu. The collapsed gibbs sampler in bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, 89(427):958–966, 1994.

MIDI. Midi tuning standard. `https://en.wikipedia.org/wiki/MIDI_Tuning_Standard`. Accessed: 2016-11-12.

Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks, 2015. URL `https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html`.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. *arXiv preprint arXiv:1604.07379*, 2016.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

Paul Smolensky. Information processing in dynamical systems: Foundations of harmony theory. Technical report, DTIC Document, 1986.

Robert H Swendsen and Jian-Sheng Wang. Nonuniversal critical dynamics in monte carlo simulations. *Physical review letters*, 58(2):86, 1987.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

Benigno Uria, Iain Murray, and Hugo Larochelle. A deep and tractable density estimator. In *ICML*, pp. 467–475, 2014.

Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, and Hugo Larochelle. Neural autoregressive distribution estimation. *arXiv preprint arXiv:1605.02226*, 2016.

Li Yao, Sherjil Ozair, Kyunghyun Cho, and Yoshua Bengio. On the equivalence between deep nade and generative stochastic networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 322–336. Springer, 2014.