Large Language Models Learn to Drum

Anonymous submission

Abstract

Automatic music generation with artificial intelligence typically requires a large amount of data which is hard to obtain for many less common genres and musical instruments. To tackle this issue, we present ongoing work and preliminary findings on the possibility for deep models to transfer knowledge from language to music, by finetuning large language models pre-trained on a massive text corpus on only hundreds of MIDI files of drum performances. We show that by doing so, one of the largest, state-of-the-art models (GPT3) is capable of generating reasonable drum grooves, while models that are not pre-trained (Transformer) shows no such ability beyond naive repetition. Evaluating generated music is a challenging task, more so is evaluating drum grooves with little precedence in literature. Hence, we propose a tailored structural evaluation method and analyze drum grooves produced by GPT3 compared to those played by human professionals, exposing the strengths and weaknesses of such generation by language-to-music transfer. Our findings suggest that language-to-music transfer learning with large language models is viable and promising.[1]

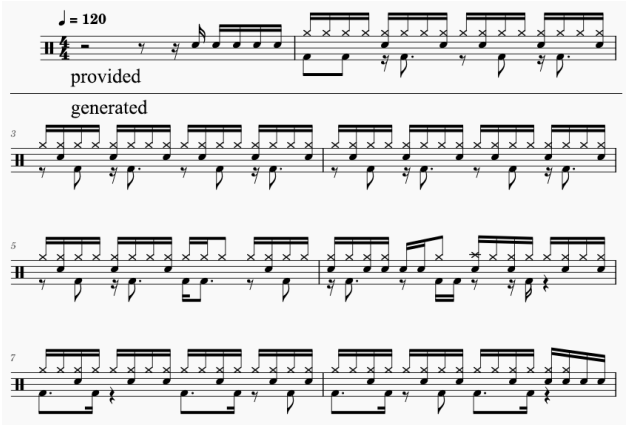


Figure 1: The sheet music of an example of our generated drum track, demonstrating our model's ability to somewhat musically follow the motif and make variations. The first two measures are provided while the rest are generated.

Introduction

Music understanding and generation using artificial intelligence has a long history (Roads 1985) and has gained steady interests in recent years (Kaliakatsos-Papakostas, Floros, and Vrahatis 2020). One strand of work focuses on symbolic music rather than audio, where music is represented as sequential data such as MIDI. While the analogy between music and language has long been studied (McMullen and Saffran 2004), the symbolic representation of music exhibits an even clearer similarity to language in their surface form. For example, music has notes, measures, and sections, while language has tokens, sentences, and paragraphs. It is thus intuitive that some work has applied natural language processing (NLP) techniques to music. Specifically, most have attempted to learn an embedding space of music (Liang et al. 2020) similar to that of texts.

Recent work has leveraged Transformers (Vaswani et al. 2017) for symbolic music processing (Huang et al. 2018; Yu et al. 2022). Transformers, when pre-trained on a massive text corpus, are known as large language models (LLMs), the current state-of-the-art approach to many NLP tasks (Devlin et al. 2019; Brown et al. 2020). Notably, Zeng et al. (2021) is one of the first and only work to pre-train a Transformer on a large symbolic music corpus containing more than 1 million songs, using a similar approach of pre-training on a text corpus, achieving state-of-the-art performance in various music understanding tasks. However, one of the most significant limitations of this and most other work in data-driven music processing is the large amount of symbolic music data required for training, which is extremely challenging to obtain for less-mainstream genres, particular styles, less-prevalent instruments, or out-of-the-ordinary specifications, severely limiting the versatility of such a method. Hence, low-resource symbolic music processing remains highly challenging. For texts, on the other hand, few-shot learning has been greatly empowered by LLMs due to the extremely large size of their pre-training textual data, magnitudes more than music data.

In this work, we are the first step to explore such **text-to-music transfer learning potential in LLMs**. In other words, we pose the hypothesis that present-day state-of-the-art LLMs, pre-trained with a massive amount of textual data, is capable of **generating symbolic music with little mu-**

[1]Data and code will be publicly released upon publication.

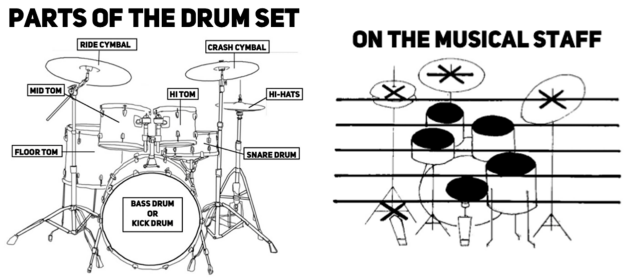


Figure 2: The anatomy of a basic drum set, and how each drum may appear on a sheet music.

	train	dev	test
Total num. MIDI	373	47	35
- rock	169	16	15
- jazz	41	6	4
- latin	37	10	3
- funk	31	6	4
- hiphop	26	1	3
- others	69	8	6

Table 1: The number of MIDI files in each style in the filtered Groove dataset used in this work.

sic data to some nontrivial extent. We specifically focus on one instrument, the drum set, for multiple reasons. First, the drum set is one of the most common and important instruments in many genres of music such as jazz, funk, blues, gospel, latin, pop, rock, metal, etc. Second, the symbolic representation of the drum set is simpler than most pitched instruments, as each note does not have a pitch but corresponds to a hit on one drum. As the number of drums is usually greatly smaller than that of possible pitches, the resulting sequence is much shorter, and thus easier to be processed by models. Third, the performance of a drum set typically is endowed with more degree of freedom with regard to the audience's aesthetics than many other instruments, making it an appropriate entry point for studying music generation with LLMs, which is presumed to be highly challenging.

We focus on the task of drum generation or composition, which has a small body of published work in literature. While most if not all all of the existing work has treated drums as an accompaniment, we instead focus on drum solo generation, with a convenient analogy to story generation at which LLMs are known to excel.

We finetune a state-of-the-art LLM, GPT3 model (Brown et al. 2020) on the Groove dataset (Gillick et al. 2019) of about 400 drum groove performances recorded as MIDI. To leverage the textual pre-training of GPT3, we propose a textual representation of a drum performance. We present the following core findings:

1. The largest and smallest GPT3 models both can generate nontrivial drum grooves after being finetuned.
2. A similar-sized model that is not pre-trained on language data, however, cannot.

We claim that the existing automatic evaluation of music generation is insufficient for our task. Hence, we propose an evaluation methodology specifically for drum grooves to both qualitatively and quantitatively evaluate the strengths and weaknesses of machine-generated drum grooves compared to those performed by humans. Finally, we provide some preliminary listening test results, with a plan to conduct scaled and rigorous tests in future work.

The Drum Set

The drum set, also known as the drum kit, or colloquially the drums, is a compound musical instrument consisting of many sub-instruments, including drums and cymbals, both generally referred to as drums in this paper. Here, we assume a simple, stereotypical drum set with a hi-hat, a crash cymbal, a ride cymbal, a bass drum, a snare drum, and a tom (see Figure 2, but without the hi-tom and the floor-tom[2]).

The performance on a drum set can be notated as sheet music, a human-readable symbolic representation, or as MIDI, which records the when each drum is hit at what velocity, a computer-readable symbolic representation.

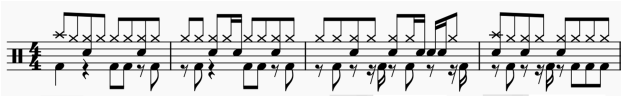
Dataset

Among just a few datasets of drum performances, Google's Groove MIDI Dataset[3] is the largest and the most high-quality to date, containing 1,150 MIDI files and over 22,000 measures of drumming by 10 professional drummers. In this dataset, the drum performances are either grooves, long sequences of rhythmic ideas, or fills, short bursts of free-flowing expressions. As we focus on drum generation or composition throughout a long sequence, we only consider the grooves in the dataset. Each MIDI is marked with the style (e.g., rock, funk, gospel, etc.), the tempo (in beats per minute, BPM), and the time signature. For simplicity, we only consider those in the time signature of 4/4. We follow the train-development-test splits in the dataset. The statistics of the filtered Groove dataset are shown in Figure 1.

The Groove dataset was originally proposed to study microtiming and expressive performance, and therefore the drum MIDI files encapsulate human imperfection. However, we re-purpose the dataset to study drum composition, leading to the following choice of simplification. First, we quantize all notes to a 16-th note grid. In other words, all note events in the MIDI are re-timed to the closest of the 16 equidistant timestamps in a measure. An implication of such quantization is that deliberate off-grid playing such as triplets or swing feels is lost. Second, we discard the velocity information (i.e., how hard a drum is hit), which can usually be inferred post-hoc to a coarse-grained extent. Third, while the drum set can be played with many expressions and articulations (e.g., hitting different part of a drum using different part of different tools), we reduce them to simply the basic articulation (head hit) of the hi-hat, crash cymbal, ride cymbal, bass drum, snare drum, and floor tom. In other words,

[2]https://i.redd.it/cen72phsihb81.png

[3]https://magenta.tensorflow.org/datasets/groove



```
1  o--o--   17 ----o-   33 ----o-   49 o-o---
2  ------   18 ------   34 ------   50 ------
3  ----o-   19 ---oo-   35 ---oo-   51 ---oo-
4  ------   20 ------   36 ------   52 ------
5  --o-o-   21 --o-o-   37 --o-o-   53 --o-o-
6  ------   22 ------   38 ------   54 ------
7  ----o-   23 ----o-   39 ----o-   55 ----o-
8  ------   24 --o---   40 ---o--   56 ---o--
9  ---oo-   25 ---oo-   41 --o-o-   57 --o-o-
10 ------   26 ------   42 ------   58 ------
11 ---oo-   27 ---oo-   43 ---oo-   59 ---oo-
12 ------   28 ------   44 --o---   60 ------
13 --o-o-   29 --o-o-   45 --o---   61 ---oo-
14 ------   30 ------   46 --o---   62 ------
15 ---oo-   31 ---oo-   47 ----o-   63 ---oo-
16 ------   32 ------   48 ---o--   64 ------
```

Figure 3: The drumroll representation of an example drum sheet music. Each measure corresponds to 16 lines in texts, where each line corresponds to a 16-th note. Each line contains 6 characters corresponding to 6 drums, where 'o' and '-' denotes whether each drum is hit.

each note only has 6 possible values. Fourth, we truncate each MIDI file to only the first 16 measures; at 128 BPM, for example, this equates to 30 seconds. Finally, we remove empty leading measures whose first quarter note is a rest, and ignore grooves with less than 8 measures.

Experiments

Representation

As discussed before, LLMs have demonstrated extremely strong few-shot transfer learning ability from one textual task to other. To possibly exploit this in music, it is necessary to come up with a textual representation of the drum grooves. We propose a pianoroll-like representation (Brunner et al. 2018), referred to as a *drumroll*, that is essentially a multi-line string where each row corresponds to a 16-th note in the time sequence, and each character in a line corresponds to whether a drum is hit. Specifically, each character is 'o' if the particular drum is hit at the particular 16-th note, and '-' otherwise. See Figure 3 for an example. To help LLMs identify the boundary between measures, we add a newline of "SEP" between every 16 lines (a measure) and a newline of "END" after the final line.

Task

We focus on an instance of drum generation referred to as drum completion, where the model is given the first 2 measures and must complete the rest of the 14 measures of the groove. The model may terminate at any point. This is analogous to conditioned story generation in NLP.

Model

First, we consider two naive baselines, **random**ly choosing whether to play a note and **repeat**ing the second given measure. We then finetune a state-of-the-art LLM, OpenAI's **GPT3 Davinci** `text-davinci-002` with 175 billion parameters, on the training set. In each file of a drum groove, the input (prompt) is the first 2 measures and the output (completion) is the remaining 14 measures. The temperature is set to 0.85 to encourage creativity. Finetuning the model on the training set costs $38.33 and takes around 30 minutes using OpenAI's API.

To ascertain the role model size plays in drum generation, we further consider a smaller **GPT3 Ada** model `text-ada-001` with 350 million parameters, which has been pre-trained on the same corpus and we use the identical settings as the larger Davinci model. Finetuning the model on the training set costs $0.51 and takes around 5 minutes using OpenAI's API. Both GPT3 models are later found to be able to generate nontrivial drum grooves.

The ascertain the role language pre-training plays in drum generation, we set the control to be an un-pre-trained GPT3 model, namely a Transformer (Vaswani et al. 2017) with the same size as GPT3. Because our computing resources cannot accommodate a model as big as Davinci, and also because merely 373 text files each with 256 lines in the training set are likely insufficient to converge the training loss, we finetune a smaller, un-pre-trained Transformer with 85 million parameters, the same magnitude as GPT3 Ada. While the training loss does converge, the model predicts the same certain sequence regardless of what 2 measures are provided, performing no better than the random baseline. This suggests that language pre-training is a necessary condition for effective drum generation.

Evaluation

How good are the drum grooves generated by GPT3? We follow the convention of music generation and consider both objective/automatic and subjective/human evaluation. We report our findings based on the test set.

Objective Evaluation

We deem the established methods of automatic evaluation of symbolic music generation unsuitable for our task. To name a few: perplexity, for example, has long been shown to exhibit low correlation with human perception in NLP (Kuribayashi et al. 2021); structural similarity, while a reasonable metric, is often approximated with some simple similarity measures (Yu et al. 2022) and discourages creativity that stray from the reference, which by no means should be treated as the ground-truth.

In contrast, we consider what constitutes a good drum groove and propose a structural evaluation called the **pattern and fill analysis**. We assume that often a good drum groove minimally satisfies the following criteria:

1. There exists one or more consistent **patterns** of some rhythmic idea and occasional change-ups known as **fills**.
2. The measures in a pattern are sufficiently similar, but ideally not identical.
3. The measures in a fill are sufficiently different from those in adjacent patterns.

Later, we verify that the human-performed grooves mostly satisfy these criteria.

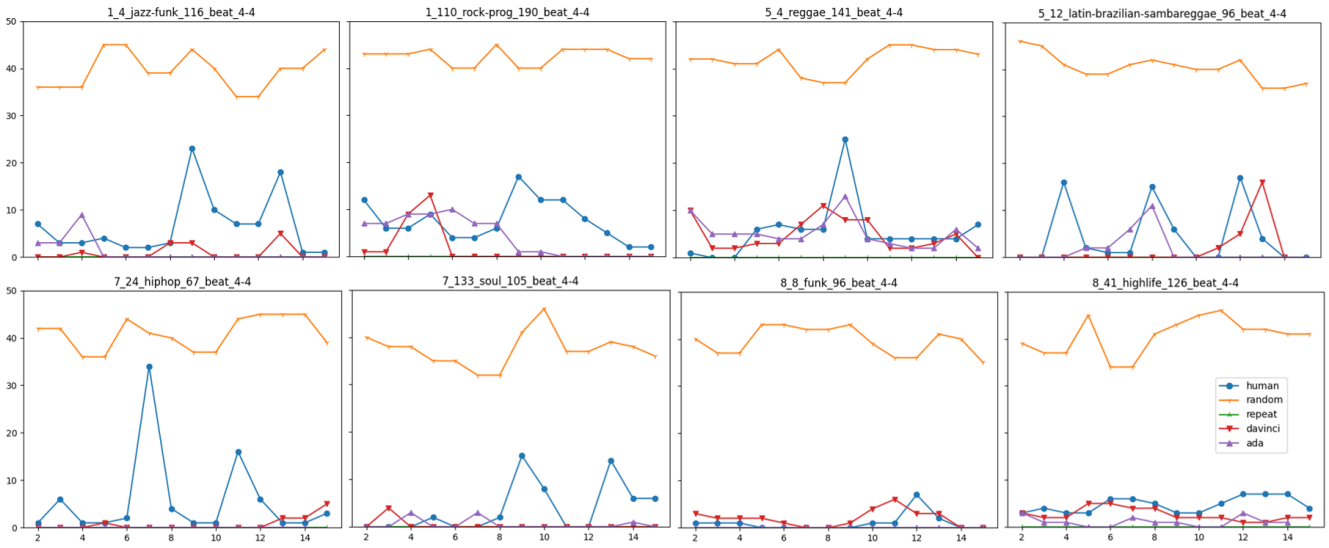


Figure 4: The variation (y-axis) of each measure (x-axis) in 8 randomly sampled drum grooves from each style.

	human	random	repeat	Davinci	Ada
avg. variation	5.1	40.4	0	3.0	3.4
avg. intra-centroids	10.1	5.7	0	6.9	7.2
avg. inter-centroids	1.4	1.2	0	0.6	0.8

Table 2: Qualitative statistics of drum grooves produced with different means. The distance between cluster centroids suggests how dissimilar the patterns and the fills are. The distance between a cluster centroid and a measure suggests the amount that the measures in one class vary by.

	human	random	repeat	Davinci	Ada
Repetitive	0	0	35	3	7
Consistent	32	0	0	29	15
Chaotic	3	35	0	3	13
Has fill	30	0	0	13	10
Avg. length	13.3	16	16	13.7	12.5

Table 3: The number of drum grooves judged to satisfy each criteria produced by each model in the test set.

In a drum performance represented as a drumroll, each measure is represented by a string of 16 lines (Figure 3). To classify each measure as either a pattern or a fill, we take a sliding window of size 3 centered at some measure m_i and calculate the edit distances between this measure and its two neighbors. The minimum of these two distances is referred to as the *variation* of the central measure:

$$\textit{variation}(m_i) = \min(dist(m_i, m_{i-1}), dist(m_i, m_{i+1}))$$

Intuitively, the variation of a measure in a pattern would be small, while that in a fill would be large. Therefore, in a good drum groove, the variation of all measures (except the first and the last) can be expected to be clearly separated.

Next, we plot the variation of each measure sequentially for each drum groove in the development set and observe consistent patterns. Eight randomly chosen grooves of different styles are shown in Figure 4. Intuitively, for human performances, the variation of a measure in a pattern should be small, while that in a fill should be large. This is indeed the case for most examples, where human-played patterns are consistent but with some variations (plateaus), while fills are largely different (spikes). As expected, the random baseline results in high variation across all measures, while the repeat baseline no variation at all – both are undesirable.

Upon qualitative examination, the two GPT3 models of different sizes can clearly generate nontrivial drum grooves, with many plateaus and occasional spikes, though less spikes than those performed by human. Quantitatively, we calculate the average variation of all measures in all grooves. As shown in Table 2, drum grooves generated by GPT3 models tend to vary less than human. To see if the generated patterns have less variation while the fills have much more variation, we perform K-means to separate the measures in a drum groove into two clusters by their variation. We then calculate the average intra-distance between the two centroids, and the average inter-distance between each measure and the centroid it is assigned to. As shown in Table 2, the intra-centroid distance shows that the grooves performed by human have a much clearer-cut pattern-versus-fill separation than GPT3, than the random baseline. The inter-centroid distance shows that the spread of variations within the class of pattern or fill is more pronounced in human-performed grooves than in GPT3-generated ones.

Subjective Evaluation

Our objective evaluation is clearly informative but insufficient. Hence, we conduct a listening test and perform an er-

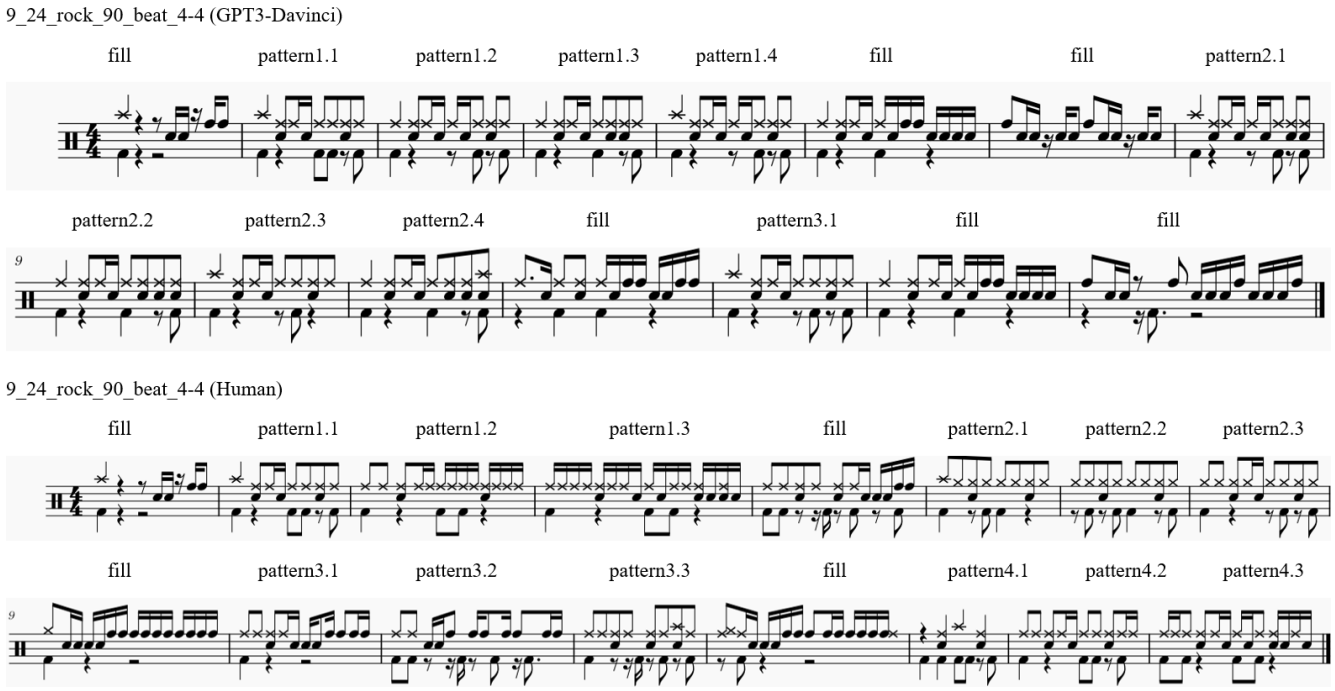


Figure 5: The sheet music of a satisfactory drum groove generated by GPT3 (above), juxtaposed with the groove with the same first 2 measures played by human from the dataset (below). We annotate each measure as either a pattern or a fill. For example, pattern$x.a$ and pattern$x.b$ are adjacent, have the same accented back-beats, but are not identical.

ror analysis based on the following criteria:

- Is the groove repetitive, meaning there is little or no variation among measures?
- Is the groove consistent, meaning there is some variation among measures but a steady rhythmic idea (specifically, the back-beat placement) can be followed?
- Is the groove chaotic, meaning there is either too much variation, or a lack of a clear rhythmic idea?
- Does the groove contain any reasonable drum fill?

While scaling up this analysis rigorously with carefully chosen subjects is left for future work, our own judgements are shown in Table 2 as preliminary findings. Concretely, all drum grooves produced via different means are shuffled and randomly present to one of the authors who has had years of training in drumming. Naturally, all randomly generated grooves are judged as chaotic without any consistent motif, while all repeated ones are by nature repetitive. For the grooves performed by human, most are judged as consistent and most include at least one fill which is sufficiently different from the rest of the rhythmic patterns. In comparison, grooves generated by GPT3 Davinci is only slightly less consistent with desirable variations among measures, but significantly less of them contain any fills, rendering the grooves less interesting and more predictable overall. For the smaller GPT3 Ada model, the observation holds to a larger extent, with more inconsistent grooves and less fills.

Case Study and Takeaways

We claim that the LLMs demonstrate an impressive ability to write good drum grooves given only hundreds of training examples without any knowledge of what features to pay attention to. From our objective and subjective evaluation above, we have observed that both LLMs and human are able to compose drum grooves with a structure and some variations. Why, however, are models' drum composition still worse than human's? We postulate several factors.

We observe that a common source of the lack of musicality in machine-generated drum grooves stems from the misplacement of back-beats, which are steadily accented beats in a measure, usually the 2nd and the 4th in 4/4 in many genres of music. Human drummers, when playing variations, tend to respect the back-beat placements, while models tend to disregard such concept. Another source of the lack of musicality is the lack of drum fills. While human drummers often inherently think about the cycle between patterns and fills to give a captivating performance, models are yet to realize the importance of those occasional deviations.

To qualitatively examine these claims, we showcase two sets drum grooves composed by GPT3, one showcasing its strengths and one its weaknesses, along with the human performed grooves with the same first 2 measures.

A positive example is shown in Figure 5. Clearly, both grooves produced by human and GPT3 alike contain consistent patterns interspersed with fills, a prototypical structure that is expected and desired by most audience. Closely examining the pattern groupings, both grooves vary the place-

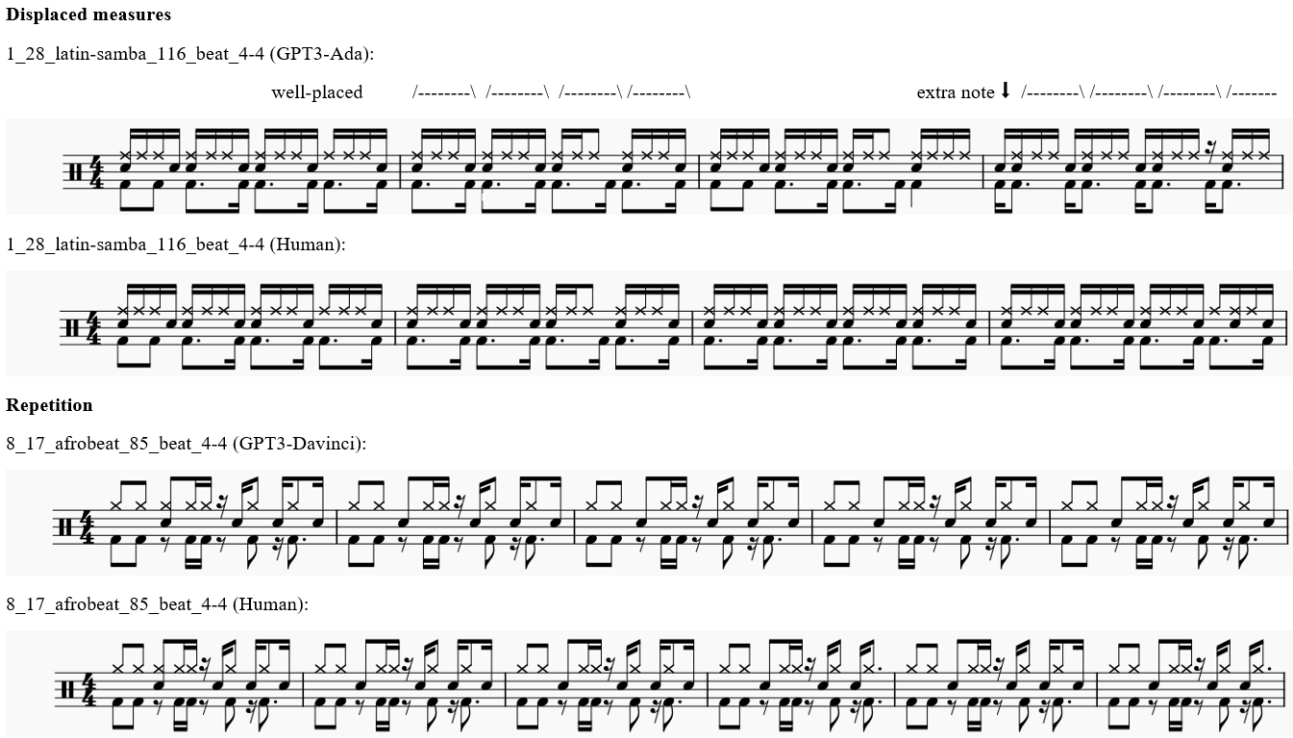


Figure 6: The sheet music of two problematic drum grooves generated by GPT3 (the 1st and 3rd), juxtaposed with those performed by human (the 2nd and 4th). In the first case, GPT3 generates an extra note at the beginning of the fourth measure, displacing all grooves afterwards. In the second case, GPT3 generates strictly repeated measures; the rest are omitted.

ment of the bass drum and the snare drum, known as syncopation, while keeping the back-beats at the 2nd and 4th quarter note in tact. As a typical drum groove, patterns in both grooves consistently place a snare drum hit in all patterns (with two exceptions in human's pattern 3.1 and 3.2). Such composition is highly desirable, as the steady back-beat placement guides the listener with a solid rhythmic foundation, while the variations keep the groove interesting and natural. Besides the ability to conforming to human drummers' standard practice, this example also showcases LLMs' ability to be creative. While the human-performed groove closely adheres to the template of 3 measures of patterns followed by 1 measure of fill, the GPT3-generated groove does not. Instead, it employs multi-measure fills, and more interestingly, grouping of an odd number of patterns followed by fills, which is uncommon in the genre.

Two negative examples are shown in Figure 6, representing two common problems of GPT3-generated drum grooves. This first case is referred to as *displaced measures*. When human drummers play a pattern, they often fit each measure with some rhythmic idea. In other words, the boundaries between two measures are often clear. In our drumroll representation, each measure, 16 lines of texts, is followed by a newline of 'SEP' to denote such boundary. However, GPT3 Ada sometimes fail to respect these boundaries and displaces the measures, thus having more "chaotic" patterns as reported in Table 3. In comparison, GPT3 Davinci makes a lot less such mistakes. The second case demonstrates LLMs' known drawback of the proclivity to repeat generations. In the given example, GPT3 Davinci generates 14 identical measures, while a human drummer would "sneak in" minor variations while maintaining the motif. It is worth noting that the repetition of drum grooves is accepted in some music genres such as pop, dance, or rock, but frowned upon in most others as they are often thought to be mechanical. For drum grooves generated by LLMs, repetition is logically a trait to be avoided.

As of now, we have taken a model-free approach where the LLMs are only trained on some drum groove data without any additional information or priors. To reinforce the strengths and alleviate the weaknesses discussed before, we suggest future work to take a modular approach instead of tackling the task in an end-to-end fashion. For example, the patterns and fills can be generated from different distributions or by different models; repetition can be explicitly discouraged by over-generating each measure and perform some voting or selection. Furthermore, it is possible to control the variation within a drum groove by injecting additional labels in the training data to condition the LLM on.

Improvisation or Recitation?

By far, we have hinted at LLMs' ability to be creative with regard to drum composition. Here, we pose one additional question: are the LLMs really *creating* some drum grooves that they have never seen during finetuning, or are they simply *regurgitating* what they have already seen?

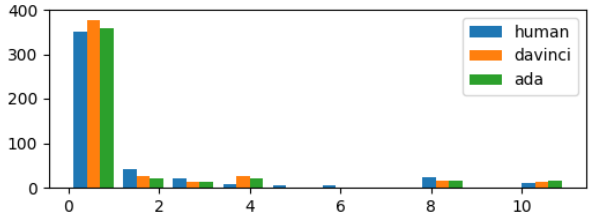


Figure 7: The count (y-axis) of occurrences (x-axis) of each generated measure appearing in the training set.

We answer this question by calculating the frequency of each generated measure in the test set appearing in the the training set. As shown in Figure 7, only a small portion of generated measures are duplicates of any seen measures during finetuning, suggesting LLMs' ability to compose novel and unseen drum grooves.

Related Work

Automatic music generation has a long history, and recent work has focused on using artificial intelligence (Kaliakatsos-Papakostas, Floros, and Vrahatis 2020) and specifically deep neural networks. Such efforts are driven by not only the prospect of having AI aid or replace human composers and arrangers for a variety of music, but also the pursuit of probing the artistic creativity of the state-of-the-art data-driven models. Most modern work in automatic music generation leverages model architectures that have shown to be effective in computer vision or NLP, such as LSTM (Lyu et al. 2015), Transformers (Huang et al. 2018; Zeng et al. 2021; Yu et al. 2022), or custom architectures to learn an embedding space (Liang et al. 2020). However, the majority of the work on music generation with AI has happened in a supervised setting, which greatly limits its application to lesser known genres or specific instruments, such as drum generation. While there is yet to be a pre-trained music generation model as versatile as its counterpart in NLP such as GPT3, we believe the exploration of language-to-music transfer is necessary but uncharted.

Automatic drum generation has a small body of existing work. Similar to music generation in general, the drum generation task can happen in various settings. In the simplest setting, only one measure of drum pattern (also known as a "beat") is generated that is supposed to repeat throughout a song (Vogl and Knees 2017; Bruford, McDonald, and Sandler 2020; Tikhonov. and Yamshchikov. 2021), while we focus on a more involved setting of non-repetitive drum composition. Alternatively, some work simply considers the general rhythm (Lattner and Grachten 2019), but not the orchestration of different drums that we emphasize on. In more practical settings, a long sequence of drum composition is generated conditioned on musical signals such as the basslines (Makris et al. 2017, 2019). While this line of work is most similar to ours which by far only deal with drum solo performance, all the work above has used drum composition data with limited size and variations as well as models (such as LSTM) that are relatively outdated in the AI community. Nevertheless, a direct comparison would be beneficial in future work. Less related is another line of work focuses on the microtiming and humanization of drum performance (Gillick et al. 2019; Burloiu and Unatc 2020; Burloiu 2020), striving to mimic human's expressive imperfections.

A small body of work has focused on rhythm games (Donahue, Lipton, and McAuley 2017; Liang, Li, and Ikeda 2019). While rhythm games and drumming have certain similarities such as the focus on note placement with regard to the rhythm, their core difference lies in that the choreography of rhythm games is optimized for difficulty and playability, while the composition of drums is optimized for musicality. Due to this fundamental difference of motivation, we consider this line of work to be mostly irrelevant to ours.

Large language models (LLMs) are deep neural models, such as Transformers, pre-trained on a massive text corpus. For example, GPT3 is pre-trained on the compilation of Common Crawl, containing most texts in the world wide net, publicly available books, Wikipedia, and so on. From BERT (Devlin et al. 2019) to GPT3 (Brown et al. 2020), LLMs have been dominant is most tasks and applications in NLP. While much about how LLMs work is unknown, and thus LLMs are notoriously known as black-boxes, there is a generally consensus that LLMs' power can be attribute to the large size of both the pre-training data and the model, which give rise to LLMs' ability to effectively adapt to low-resource domains via transfer learning by being finetuned on a small amount of data. Interestingly, a few recent work has found that some of these abilities include transfer from pre-trained language to non-language tasks, such as chess (Stöckl 2021). While non-language textual representations such as chess moves or music charts are similar to natural language superficially, each manifests vastly different structures, and so we claim that transfer learning between them is well worth studying.

Conclusion and Future Work

Our preliminary findings show that pre-trained large language models (LLMs) finetuned with merely hundreds of symbolic music files, such as drum grooves, can learn to generate music non-trivially. We also provide evidence that such ability can be attributed both the model size and the presence of language pre-training. We hope this observation inspires research efforts in not only low-resource music generation, but also exploration of extraordinary potentials of LLMs. We also attempt to pioneer the automatic evaluation of drum grooves which we hope to facilitate future work in drum generation with AI.

We also briefly discuss our plans for future work. While drum generation is scarce in literature and challenging to reproduce, we will still strive compare some existing specialized models. While our current drumroll representation ignores velocity, such information can easily be encoded by replacing the marker 'o' with the velocity value. However, the effects of doing so remains to be explored. Our methodology might be ported to other instruments such as piano, which we plan to explore, whereas it would be more involved to tackle multi-instrument conditioned generation.

References

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 1877–1901. Curran Associates, Inc.

Bruford, F.; McDonald, S.; and Sandler, M. 2020. jaki: user-controllable generation of drum patterns using an LSTM encoder-decoder and deep reinforcement learning. In *The 2020 Joint Conference on AI Music Creativity*.

Brunner, G.; Wang, Y.; Wattenhofer, R.; and Zhao, S. 2018. Symbolic music genre transfer with cyclegan. In *2018 ieee 30th international conference on tools with artificial intelligence (ictai)*, 786–793. IEEE.

Burloiu, G. 2020. Interactive Learning of Microtiming in an Expressive Drum Machine. In *The 2020 Joint Conference on AI Music Creativity*.

Burloiu, G.; and Unatc, C. 2020. Adaptive Drum Machine Microtiming with Transfer Learning and RNNs. In *Extended Abstracts for the Late-Breaking Demo Session of the International Society for Music Information Retrieval Conference (ISMIR)*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Donahue, C.; Lipton, Z. C.; and McAuley, J. 2017. Dance dance convolution. In *International conference on machine learning*, 1039–1048. PMLR.

Gillick, J.; Roberts, A.; Engel, J.; Eck, D.; and Bamman, D. 2019. Learning to groove with inverse sequence transformations. In *International Conference on Machine Learning*, 2269–2279. PMLR.

Huang, C.-Z. A.; Vaswani, A.; Uszkoreit, J.; Shazeer, N.; Simon, I.; Hawthorne, C.; Dai, A. M.; Hoffman, M. D.; Dinculescu, M.; and Eck, D. 2018. Music transformer. *arXiv preprint arXiv:1809.04281*.

Kaliakatsos-Papakostas, M.; Floros, A.; and Vrahatis, M. N. 2020. Artificial intelligence methods for music generation: a review and future perspectives. *Nature-Inspired Computation and Swarm Intelligence*, 217–245.

Kuribayashi, T.; Oseki, Y.; Ito, T.; Yoshida, R.; Asahara, M.; and Inui, K. 2021. Lower Perplexity is Not Always Human-Like. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 5203–5217. Online: Association for Computational Linguistics.

Lattner, S.; and Grachten, M. 2019. High-level control of drum track generation using learned patterns of rhythmic interaction. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 35–39. IEEE.

Liang, H.; Lei, W.; Chan, P. Y.; Yang, Z.; Sun, M.; and Chua, T.-S. 2020. Pirhdy: Learning pitch-, rhythm-, and dynamics-aware embeddings for symbolic music. In *Proceedings of the 28th ACM International Conference on Multimedia*, 574–582.

Liang, Y.; Li, W.; and Ikeda, K. 2019. Procedural content generation of rhythm games using deep learning methods. In *Joint International Conference on Entertainment Computing and Serious Games*, 134–145. Springer.

Lyu, Q.; Wu, Z.; Zhu, J.; and Meng, H. 2015. Modelling high-dimensional sequences with lstm-rtrbm: Application to polyphonic music generation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Makris, D.; Kaliakatsos-Papakostas, M.; Karydis, I.; and Kermanidis, K. L. 2017. Combining LSTM and feed forward neural networks for conditional rhythm composition. In *International conference on engineering applications of neural networks*, 570–582. Springer.

Makris, D.; Kaliakatsos-Papakostas, M.; Karydis, I.; and Kermanidis, K. L. 2019. Conditional neural sequence learners for generating drums' rhythms. *Neural Computing and Applications*, 31(6): 1793–1804.

McMullen, E.; and Saffran, J. R. 2004. Music and Language: A Developmental Comparison. *Music Perception: An Interdisciplinary Journal*, 21(3): 289–311.

Roads, C. 1985. Research in music and artificial intelligence. *ACM Computing Surveys (CSUR)*, 17(2): 163–190.

Stöckl, A. 2021. Watching a Language Model Learning Chess. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, 1369–1379. Held Online: INCOMA Ltd.

Tikhonov., A.; and Yamshchikov., I. 2021. Artificial Neural Networks Jamming on the Beat. In *Proceedings of the 6th International Conference on Complexity, Future Information Systems and Risk - COMPLEXIS,*, 37–44. INSTICC, SciTePress. ISBN 978-989-758-505-0.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Vogl, R.; and Knees, P. 2017. An intelligent drum machine for electronic dance music production and performance. In *NIME*, 251–256.

Yu, B.; Lu, P.; Wang, R.; Hu, W.; Tan, X.; Ye, W.; Zhang, S.; Qin, T.; and Liu, T.-Y. 2022. Museformer: Transformer with Fine- and Coarse-Grained Attention for Music Generation.

Zeng, M.; Tan, X.; Wang, R.; Ju, Z.; Qin, T.; and Liu, T.-Y. 2021. Musicbert: Symbolic music understanding with large-scale pre-training. *arXiv preprint arXiv:2106.05630*.