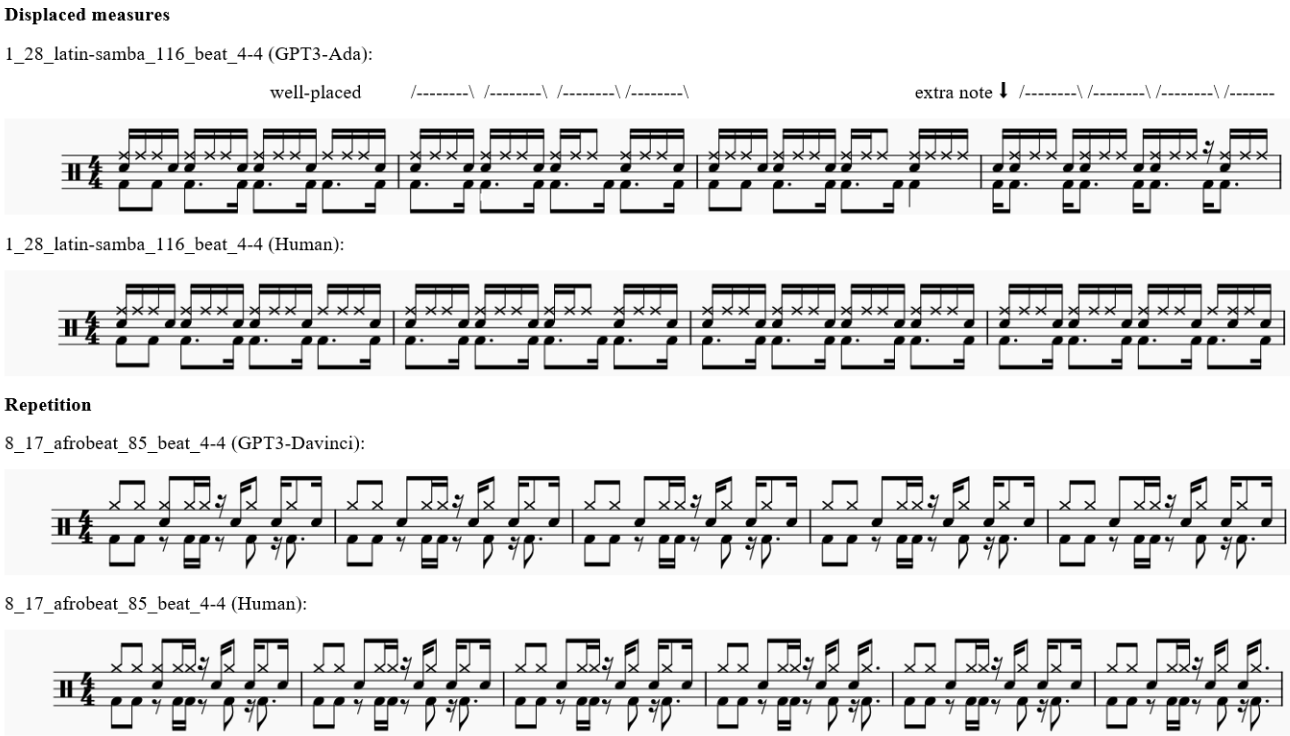

Figure 6: The sheet music of two problematic drum grooves generated by GPT3 (the 1st and 3rd), juxtaposed with those performed by human (the 2nd and 4th). In the first case, GPT3 generates an extra note at the beginning of the fourth measure, displacing all grooves afterwards. In the second case, GPT3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

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?

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.

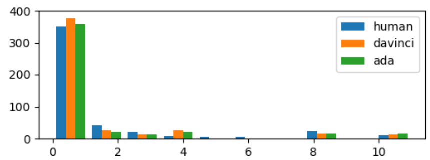

Figure 7: The count (y-axis) of occurrences (x-axis) of each generated measure appearing in the training set.

Related Work

Automatic music generation has a long history, and recent work has focused on using artificial intelligence (**?**) 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 (**?**), Transformers (**???**), or custom architectures to learn an embedding space (**?**). 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 (**???**), while we focus on a more involved setting of non-repetitive drum composition. Alternatively, some work simply considers the general rhythm (**?**), 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 (**??**). 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 (**???**), striving to mimic human's expressive imperfections.

A small body of work has focused on rhythm games (**??**). 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 (**?**) to GPT3 (**?**), 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 (**?**). 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.

Porro maxime autem, corrupti excepturi quaerat veritatis odit, sunt minima iure molestias non asperiores repudiandae inventore sint, pariatur doloribus reprehenderit fugit impedit laudantium?Soluta saepe tenetur perferendis alias dolorem corporis explicabo illo ad id, temporibus eius ab, alias iusto itaque deleniti deserunt unde similique non ducimus aliquid dicta provident?Alias inventore nesciunt, harum sunt ipsam?Minus praesentium accusantium explicabo commodi veniam quis quisquam, hic rerum amet, reprehenderit inventore nobis tenetur voluptate consectetur eaque facere perferendis, unde deserunt officia est tempora sint ex placeat, vel rerum assumenda commodi aut exercitationem officiis unde illum illo accusantium?Totam expedita corrupti porro nobis ratione aliquid asperiores quasi aliquam esse eligendi, ipsa architecto amet fuga quam asperiores esse fugiat quia non vitae quidem, quisquam architecto debitis dolores quae dignissimos magnam at possimus pariatur distinctio?Unde vitae iure a architecto nemo aliquid harum totam, ut id harum