DALARNA UNIVERSITY

Degree Thesis

Master's level(Second cycle)

Machines Learning from Other Machines?

Implications of Model Consistency on Future Large Language Models Learning from AI-Generated Content

Authors: Balaji Vijayaraj, Satheendra Liyanagunawardena
School: Dalarna University
Supervisor: Arend Hintze
Examiner: Moudud Alam
Subject/main field of study: Microdata Analysis
Course code: MI4001
Credits: 30
Date of examination: 3rd June 2024

At Dalarna University it is possible to publish the student thesis in full text in DiVA. The publishing is Open Access, which means the work will be freely accessible to read and download on the internet. This will significantly increase the dissemination and visibility of the student thesis.

Open Access is becoming the standard route for spreading scientific and academic information on the internet. Dalarna University recommends that both researchers as well as students publish their work Open Access.

I give my/we give our consent for full text publishing (freely accessible on the internet, Open Access):

Yes ☒ No ☐

Abstract

The recent advent of generative AI tools - specifically large language models, populate the internet with increasing amounts of AI-generated content. While these tools based on large transformer architectures are originally trained on data from the content of the internet itself, there is little knowledge about the effects of AI-generated content entering their own training dataset. This thesis investigates how AI-generated content within training datasets may affect the training and performance of Large Language Models (LLMs). Using a scaled-down transformer model, it explores how sequential learning, data overlap, and contradictions influence model accuracy and reliability. Experimentally, the study evaluates the prediction accuracy decline and robustness in transformer models when they sequentially learn from each other's predicted output, using a training dataset of character sequences which overlap and contradict with each other at three different levels (none, moderate, high). The findings highlight that models' prediction accuracy decreases over time when their own predicted output enters the training dataset. It further showed that models learn better and sustain sequential prediction accuracy longer when there is some overlapping information in the training data, as opposed to having no overlapping information. Moreover, the study highlighted that transformer models with more embedding layers, hidden dimension and embedding size learn with higher accuracy compared to those with fewer. Overall, by using scaled-down transformer models on various sequence datasets, this paper provides insights into challenges that current and future LLMs may face - declining prediction accuracy when trained on their own generated content, less robustness to overlapping and contradictory training data, and therefore less reliability over time. The research, however, is limited by the use of scaled-down models, which may not capture the full capabilities and performance of larger and more sophisticated models used in practical applications. Hence, further research with more sophisticated models and complex datasets is necessary to support these findings fully.

Keywords: Large Language Models (LLMs), Transformers, Sequential Learning, Data Overlap, Data Contradiction, Model Size, Data Complexity, AI-generated Content, Training Data Pollution.

Acknowledgement

We would like to extend our deepest gratitude to our supervisor, Professor Arend Hintze, for his invaluable guidance and unwavering support throughout the course of this research. His expertise and insightful critiques have been instrumental in shaping both the direction and the outcome of this study. Additionally, we acknowledge the use of ChatGPT, an AI language tool developed by OpenAI, in refining grammar for better coherence and clarity of certain sections of this report. We take full responsibility for the content of this report and confirm that the refined texts do convey our originally intended ideas.

Contents

1 Introduction

1.1 All eyes on 'AI tools'

The advent of Artificial Intelligence (AI) has been propelled by significant breakthroughs in machine learning, particularly in the domain of deep learning [1]. Deep learning, facilitated by big data and robust computing, has led to highly advanced AI models capable of learning from complex datasets and making decisions with remarkable accuracy. In the recent years, the world has shifted its attention towards such AI-driven tools, particularly Generative AI models.

Large Language Models (LLMs) are a type of Generative AI model which excel at natural language understanding and generation tasks, including language translation, text summarization, and content generation [2]. In parallel to GPT-3.5, GPT-4 and GPT-4o introduced by Open AI[3], Meta introduced Llamaa 3 [4], Anthropic introduced Claude 3[5], and Google introduced BARD[6] and then Gemini 1.5 [7]. Having trained on vast amounts of raw data from the internet, they can now produce remarkably realistic text outputs that are difficult to distinguish from human-generated content [8, 9].

1.2 Transformers

The aforementioned LLMs share the same building block - the transformer architecture. Transformers have changed the way we handle complex tasks, including natural language processing (NLP). The self-attention mechanism is the heart of the Transformer. This mechanism enables the model to prioritize the significance of individual words within a sentence, irrespective of their position [10]. Built on such an architecture and design to effectively process sequential data within neural networks, LLMs are skilled in statistically identifying the most probable next token in a sequence of tokens[11]. This functionality exists in a model regardless of its scale. Therefore, in principle, the implications of a small-scale transformer model should be transferable to large-scale Language Models.

Transformers are versatile and find applications not only in natural language processing but also in domains such as image recognition and audio processing. Advanced models such as BERT, GPT, and T5 are each tailored to specific tasks for both research and practical implementations. Their ability to handle multiple tasks simultaneously makes them highly efficient and scalable. For example, GPT-3 has been pre-trained on a text dataset of 45 terabytes of text data resulting in approximately 175 billion parameters[12]. BERT, another popular LLM trained on an unspecified data scale has around 340 million parameters [13].

1.3 Content pollution in the era of content generation?

Given their rapid advancements, our environment becomes increasingly more populated by content generated by these machines [14, 15, 16, 17, 18, 19]. the majority of their training data are content freely available on the internet, aside from third party data. It is projected that by 2026, the current supply of data available to train large models may be exhausted [20]. Therefore, it may be inevitable that when such a model is trained in the future, its training data set might intentionally or accidentally include content that itself or other AI tools created [21, 22].

Recently, concerns have emerged regarding misinformation generation, factual errors, source fabrication, and malicious content [15, 14, 18]. While in most platforms, the current remedies for handling such errors include traditional techniques such as hand-curating content [23, 24, 25, 26, 27], it becomes increasingly challenging to keep up with the rate of AI content generation and identifying [28] or removing them will not be straightforward [21, 29, 30] or practical [31, 32].

Generative AI can amplify the spread of misinformation when people unknowingly trust the accuracy of the content it produces [12, 17, 16, 14, 33]. Moreover, the negative social impact of AI generated content can span from privacy risks to escalating social inequality and therefore must not be negated [34, 35, 36, 37]. However, exploring their behaviour to these issues at the heart of the problem - by testing on the large models themselves is out of reach for the majority of the research community. The amount of computing resources, training time, and having billions of parameters has led to only industry working with such AI models [38]. From the consumers' perspective, the impact of content pollution becomes increasingly apparent. Given such current circumstances, what would happen if AI generated content become part of their own training data? The question awaits us at the end of the tunnel [39, 22, 40].

1.4 Overlap and Contradiction

Apart from content pollution, there is the added dimension of overlap and contradiction present in a training dataset of a large generative model. Within the text dataset of 45 terabytes used to train ChatGPT, for example, it is obvious that there is ample possibility for it to contain overlaps as well as contradictions. The trained model should ideally not be biased towards one concept, however, some studies reveal insights into its potential predisposition towards certain ideologies [41].

For example, consider liberalism and social democracy. Both doctrines share similar and overlapping sentiments towards democracy and protecting individual rights. Additionally, they agree that the government should play a role in the economy to help protect the welfare of its citizens. This is an example of overlapping ideas.

Capitalism and socialism are two contrasting ideologies regarding how economies and

societies should be structured. In capitalism, there is a focus on private property rights and allowing the market to operate freely to encourage business growth. On the other hand, socialism suggests that resources should be shared among everyone, aiming for a fairer society without private ownership.

Imagine training a model on ideas that both overlap and contradict each other. What would it learn better? Can such a model grasp both types of ideas equally well, or does it lean towards one ideology over the other?

To answer these questions, it is important to understand how specific data with overlap and contradiction affect the learning ability of machine learning models.

1.5 Model Size and Data Complexity

Consider also the size of the model. Does a larger model manage to capture the essence of these ideas more effectively than a smaller one? Furthermore, think about the complexity of the training data. How might the intricacies of the data affect the learning outcomes of the model?

The behaviour of data overlap, contradiction, and the effect of content degradation over time may vary with model size and complexity. Using sequential data to train a scaled-down version of a transformer model and interpreting the results, we aim to explore possible future implications of these effects on large-scale language models.

1.6 What to look for?

In this paper, we investigate the impact of AI generated content entering their own training data, particularly focusing on Large Language Models (LLMs). Generative AI models such as GPT-3, Llamaa 3, Claude 3, and BARD are remarkable at generating human-like text outputs. However, the effects of AI-generated misinformation, factual errors, and overall "polluted" data raises concerns regarding the reliability and integrity of their outputs. Therefore, it need to be investigated to understand their imminent effect on LLMs in the near future. [12, 38]. Additionally, size of the model, complexity, overlap, and contradiction within training data are important factors in understanding the implications of such a future. However, exploring these facets at the heart of the problem - by testing on LLMs themselves is out of reach for the majority of the research community. The amount of computing resources, training time, and having billions of parameters has led to only industry working with LLMs [38]. While the problems described above might become an issue for large language models in the future, we have no knowledge about this phenomenon, even for smaller models.

With this in mind, our study contributes through two primary objectives:

- Impact of Sequential learning: We investigate the consequences of LLMs generating their own, potentially erroneous, output that subsequently enters their own

training dataset.

- Robustness to Contradictory and Overlapping Information: Furthermore, we explore the robustness and learning ability of LLMs against the influence of contradictory and overlapping information.

By scaling down to a simplified transformer model, we attempt to offer insight into the potential challenges awaiting large-scale language models in the foreseeable future. In the subsequent sections, we present related works, our study design and methodology, results of our findings, discussion, limitations of the study, and conclusion.

2 Literature Review

Let us now discuss concepts and works related to our study. We initiate this section by distinguishing the differences between humans learning and AI learning followed by a brief introduction to the key aspects of the transformer architecture. We then discuss the positive aspects of Generative AI, their adverse effects, and future challenges and limitations such as hallucination. Upon discussing the concepts of model size, data overlap and contradiction, we conclude this section by introducing the aims and contributions of our study to the existing literature.

2.1 Do AI models learn the same way as humans do?

Humans learn from each other through social interactions, utilizing sensory experiences, cognitive processes, and communication [42]. This involves observing, imitating, and receiving feedback, often facilitated by language[43]. Humans can also generalize knowledge across various domains, though individual learning experiences vary. Humans exhibit creative thinking, innovation, and adaptability, generating novel ideas and exploring alternative solutions[44].

Artificial intelligence learns solely from algorithms and mathematical models trained on data, lacking the ability to compare observations against the world[45]. Unlike humans, AI systems process vast amounts of data rapidly and learn from numerous examples, only enabling efficient handling of tasks at scale. While in terms of scale and speed, AI has a significant competitive advantage, the limitations of AI learning may affect the quality of their outputs, especially as more AI generated content enter training datasets. AI systems can mimic creativity to some extent but rely on learned patterns from existing data and may not achieve genuine innovation like humans[46].

However, AI models are getting closer to reaching human learning by methods such as large-scale training [2], reinforcement learning [47] with human feedback, and self-supervised learning [48]. Advances such as the Transformer architectures spearhead AI towards bridging this gap further.

2.2 Transformers revolutionizing AI

Transformer models have revolutionized the field of natural language processing (NLP) and beyond, since their introduction in the seminal paper "Attention is All You Need" by Vaswani et al. in 2017 [10]. Distinguished by their reliance on self-attention mechanisms, Transformers are designed to process data in parallel rather than sequentially, which clearly improves the efficiency of model training and opens up new possibilities for handling vast amounts of data.

At the core of the Transformer architecture is the self-attention mechanism, which allows the model to weigh the importance of different words within a sentence, regard-

less of their positional distance from each other. This feature enables the model to capture complex syntactic and semantic relationships more effectively than was possible with previous models that relied heavily on recurrent neural networks (RNNs) and convolutional neural networks (CNNs).

Transformers consist of an encoder and a decoder. The encoder processes the input data and transforms it into a continuous representation that holds all the learned insights of the input. The decoder then takes this representation and step-by-step generates the output for tasks such as translation, summarization, or text generation. Each of these components is composed of multiple layers of self-attention and position-wise, fully connected layers.

The adaptability of Transformers has been demonstrated across a variety of applications, not just in NLP but also in areas such as computer vision and audio processing [1]. The model's ability to handle parallel computations effectively makes it highly scalable and efficient, which has led to the development of various sophisticated variants such as BERT [49], GPT [2, 50, 3], and T5, each tailored for specific tasks and challenges in both academia and industry [51, 52].

The introduction of Transformer models has thus marked a pivotal shift in machine learning techniques, steering subsequent innovations towards models that prioritize efficiency, scalability, and the capacity to grasp deeper contextual meanings across different types of data.

2.3 The light at the end of the tunnel - towards bliss or abyss?

2.3.1 The positive impact of Generative AI

A large number of organizations have adopted AI assistance for decision support. These AI assistants act as intelligent question-answering systems. Used commonly in information technology (IT) help desks, they support transitional knowledge work tasks and provide solutions for mundane needs such as cooking recipes and medical advice [53]. Businesses across various sectors prepare to expect extensive benefits as integration of more generative AI tools into established software packages and technology platforms increase. The global gross domestic product is forecasted to increase by 7% while replacing 300 million jobs of knowledge workers [51]. These advantages span from boosting office efficiency and sales initiatives to refining architectural designs and manufacturing processes, to enhancing diagnostic capabilities in healthcare, and even to detecting instances of cyber fraud. One study suggests that generative AI may even help bypass the need for real data in cases such as Medical Vision-Language Pre-training [54]. Another study proposes synthetic data generated by AI may help eliminate societal and historic biases in traditional datasets [55]. They suggest that by doing so, not only will the dataset be more representative and realistic, but also alleviate privacy concerns related to existing datasets.

2.3.2 AI-generated content - A double-edged sword

However, generative AI models do not always produce accurate outputs. The reason is that machine learning models such as LLMs rely on probabilistic algorithms for making inferences. Thus, generative AI models such as GPT-4 generate the most probable response to a prompt, not necessarily the correct response. On the flipside of aforementioned studies speaking for models training on synthetic data, some studies claim that it will lead to poor performance of the trained neural networks at the inference stage [56]. This is attributed to the inability of synthetic data to capture the reality of the domain or experiment.

Following its release, StackOverflow banned answers generated or "reworded" by ChatGPT, claiming that the rate of receiving correct answers from generative AI such as ChatGPT was "too low" [57]. On one hand, content moderation becomes increasingly challenging due to the sheer rate of AI content generation [14], and on the other, it becomes increasingly difficult to differentiate AI-generated content from human-generated content. One study found that humans find it difficult to distinguish GPT-3-generated tweets from human-written tweets [58]. Several studies align with the same sentiment by observing such difficulties in various other settings [30, 28, 59, 37, 60]. Given such circumstances, there is even more potential for AI-generated language to enable novel forms of plagiarism, manipulation, and fabrication [13].

Generative AI can also amplify the spread of misinformation when people unknowingly trust the accuracy of the content it produces. One study highlights the risks of of AI creating dangerous advice as well as erroneous and nonsensical information about mental illness with fabricated sources[12]. Another article informs how it was discovered that people use ChatGPT to write product reviews in Amazon.com, due to the language model's commonly used phrase, "As an AI language model,..." in its writing[16]. One paper highlighted the potential of large generative models to both generate and detect false news commonly referred to as "fake news"[33].

2.3.3 Future outlook and challenges of training new AI

While it is projected that the currently available data pool may deplete by 2026, several entities have already begun discussions about training future AI with synthetic data. In parallel, companies like OpenAI and Google have resorted to drastic measures to start harvesting data, treading in legal gray areas [61]. Although such synthetic data may be held to some level of quality standard, the success or failure of this remains speculative [62]. Nevertheless, given the already existing AI generated content available on the internet, a new AI model's training data set might intentionally or accidentally include content that itself or other AI tools created [21, 22]. What would be the outcome of such a situation? Currently, only a handful of literature explore this question. A paper titled "Self-Consuming Generative Models Go MAD" highlights the dangers of autophagous (self-consuming)

loops using state-of-the-art image generating models [39]. The study found that using synthetic data to train next-generation models results in a phenomenon called "model collapse." The errors and biases inherent in the synthetic data (model-generated outputs) get amplified as the model learns from its own generated outputs. Thus it causes next generation model outputs to rapidly degrade and deviate significantly from the original images. Another recent paper discusses "Model Collapse", a degenerative process which affects subsequent models in the sequence [40]. The model outputs result in polluting the training set of the subsequent models. They explore this idea on Gaussian Mixture Models, Variational Autoencoders, and a fine-tuned version of the pre-trained OPT-125m causal language model made available by Meta through Huggingface. They noted two variations of model degeneration. Firstly, "early model collapse" where the model begins losing information about the tails of the distribution. Secondly, late model collapse where the model converges to a distribution deviating significantly from the original, often with very small variance. Another study conducted a similar experiment where diffusion models were trained on a mixture of real and AI-generated data. They found that the quality of the generated images deteriorate as more AI-generated data is used for training [22].

2.3.4 Hallucinations in AI: From 'Hi, How Are You?' to 'How High Are You?'

Transformer models, particularly those trained for sequence generation tasks, such as language modeling or text generation, are susceptible to hallucinatory behaviors due to their autoregressive nature and capacity to capture complex patterns in data [63]. Hallucination, within the context of transformer models, refers to the phenomenon where the model generates repetitive or patterned sequences that deviate significantly from the training data distribution while still appearing coherent and plausible at first glance [64].

In sequential learning tasks, where models are trained to generate sequences of data, the iterative nature of the training process can amplify the risk of hallucinatory behaviors. As models iteratively generate and refine sequences based on their own predictions, errors and inaccuracies may accumulate over time, leading to increasingly divergent outputs [40, 39]. One study demonstrated that as the input sequences grow in length, attention across tokens is diluted and leads to hallucinatory behaviours. Another paper [65] proves through learning theory that hallucination is an unavoidable issue in large language models (LLMs) due to inherent limitations in learning all computable functions. This paper demonstrates that even with mitigation efforts, hallucinations are inevitable both in simplified models and in real-world applications.

While studies such as the above state the inevitability of model hallucination, some studies also suggest possible approaches to mitigate it. One popular approach for improving robustness to such errors include chain-of-thought generation, in other words, explicitly providing intermediate reasoning steps to the model [66]. An analysis in the

book "Truth, Lies, and Automation" echoes a similar view - LLMs such as GPT-3 are observed to deliver better results if a complex task is segmented into sub-tasks and have them perform each in sequence. For example, as opposed to prompting it to rewrite an article, a better approach would be to initially summarize the original article in bulltet points, followed by using the bullet points to rewrite the article[67].

2.4 Model Size and Data Complexity

Consider also the size of the model. Does a larger model manage to capture the essence of these ideas more effectively than a smaller one? Furthermore, think about the complexity of the training data. How might the intricacies of the data affect the learning outcomes of the model?

Kaplan et al. [68] provided a comprehensive analysis of scaling laws specifically for neural language models. They showed that performance, measured by perplexity(inability to understanding something), follows power-law relationships with respect to model size, dataset size, and computation. This work was seminal in highlighting that, contrary to previous assumptions, model size is a critical factor in achieving lower perplexity, even more so than the amount of training data or compute used. More about the assumption of model size and data complexity in our experiment is explained in the result section.

2.5 Overlap and Contradiction

First, we need to see what is overlap and contradiction with respect to the training data (songs). For that we consider 2 songs "ABCDEFGHIJABCDEFGHIJABCDEFGHIJABC--DEFGHIJAB" and "ABCDEFGHIEABCDEFGHIEABCDEFGHIEABCDEFGHIEAB", where they differ at positions where "E" appears.

Overlap in training data occurs when identical input sequences from different songs produce the same output sequence. For example, the sequence “ABCD” in both songs leads to the output “BCDE”. Such overlaps provide consistent training signals to the model, facilitating its learning process by reinforcing predictable patterns.

Contradictions arise when the same input sequence yields different output sequences in different songs. A notable instance is the input “FGHI”, where:

- In the first song, “FGHI” leads to the output “GHIJ”.
- In the second song, the same input “FGHI” leads to “GHIE”.

The detailed method of how we calculate and assess these overlaps and contradictions is discussed in the upcoming methodology section. This will provide a clearer picture of our approach and the insights gained from using a decision tree classifier to evaluate the predictive accuracy across varying contexts of overlap and contradiction.

2.6 Objective and aim of the study

Our study aims to address two fundamental questions related to the behaviour of deep learning neural network models, specifically focusing on Transformers:

1. What are the implications of incorporating a model's own predictions, potentially erroneous, back into its training dataset?

2. How does the Transformer model's robustness vary in response to contradictory and overlapping information, and what impact do variations in model size and data complexity have on its learning capabilities?

In the next section, we discuss the study design and methodology we utilized explore these questions.

3 Methodology

In this section, we first introduce the architecture of the transformer model utilized for this study along with its specifications. We then explain the dataset chosen and the choice behind it, followed by how we designed the dataset. Then, after explaining our data preparation steps of extraction, mixing, and shuffling, we conclude this section by detailing the model training and testing processes along with the statistical metrics used for evaluating model performance.

3.1 Study Design

Our study is centered around two main categories, under which we conduct several experiments. From a broader perspective, the two main categories are:

Category 1: Experiments are designed to evaluate the effects of a model's own or other models' potentially erroneous outputs entering its training dataset. From a high-level, this experiment examines the notion of recurrent AI-generated content pollution, and their impact on a model's learning accuracy and stability.

Category 2: Experiments are focused on assessing how different scales of model, data complexity and varying degrees of overlap and contradiction in training data influence the model's ability to learn effectively.

The model specifications, dataset, training and testing setups/procedures, and calculation of statistical metrics are all adopted to align with the two aforementioned categories.

3.2 Model Architecture

We implemented a Transformer model[10] using PyTorch. The model consists of several key components:

1. **Embedding Layer:** Converts input tokens to dense vectors.
2. **Positional Encoding:** Adds positional information to embeddings.
3. **Transformer Encoder:**
    - **Self-Attention Mechanism:** The core part of the transformer that allows the model to weigh the importance of different tokens in the sequence.
    - In the model, this is handled by `nn.TransformerEncoderLayer` with multi-head attention.
    - **Feed-Forward Network:** A series of linear transformations and non-linear activations.

- This is implemented in the model by the linear layers within `nn.TransformerEncoderLayer`.
- **Normalization and Dropout:** Applied to stabilize and regularize the training process.
- These are included in the model with `LayerNorm` and `Dropout`.

4. **Decoder (Output Projection):** Maps the encoder's output to the vocabulary space to generate logits.
    - The model uses a linear layer to project the output of the transformer encoder back to the size of the vocabulary.
5. **Masking:** Ensures that predictions for a specific position depend only on the known outputs at positions before it.

3.3 Model Specifications

The model includes the following specifications:

- **ntokens:** The size of the vocabulary. Here we used 10 tokens and that is constant in all the experiments.
- **emsize:** The dimension of the embedding space. We varied this based on the model size from 4 to 16.
- **nhead:** The number of heads in the multi-head attention mechanism. we kept it constant at 2 throughout the experiment.
- **d_hid:** The dimension of the feed-forward neural network within the encoder. We varied this based on the model size from 4 to 16.
- **nlayers:** The number of stacked encoder layers. We varied this based on the model size from 1 to 4.
- **dropout:** The dropout rate used in positional encoding and encoder layers to prevent over fitting. This is kept constant at 0.03 throughout the experiment

Weights within the model are initialized uniformly in a small range and adjusted during training through back propagation.

3.4 Dataset - what exactly are "Songs" in this study?

Music adheres to the concept of time and is sequential, among other properties such as dynamics, rhythm, and timbre. Songs are often a combination of several of these properties. While in different musical cultures, the primary notes are denoted differently, in Western music, the musical notes are denoted by A, B, C, D, E, F, and G. Songs are created when some notes are played in a sequence.

This study utilizes a dataset of musical sequences, specifically starting with the song "Daisy Bell," which is a sequence of 42 notes, created using notes, A, B, C, D, E, F, G, from standard western music but also including the unconventional 'H', 'I', 'J' notes to increase complexity. While 'H', 'I', 'J' are not standard musical notes in Western music, its inclusion is strategic, aiming to enhance the dataset's complexity to challenge the model beyond simple sequences. The notes, A, B, C, D, E, F, G, H, I, J were encoded from 0 to 9[69].

Each created song was designed to loop back to its initial state, creating a cyclic pattern. While we refer to these character sequences as "songs" for the namesake, it must be noted here that they were not aimed to be representative of other real songs in popular culture or be true to western music notation. Our goal is not to analyze the music itself but to test the sequence prediction capabilities of neural networks on data with structured, time-series properties. The inclusion of 'H', 'I', 'J' helps simulate real-world unpredictability, ensuring the models learn to recognize and predict complex patterns. This approach allows us to focus on structural patterns within the data, helping us understand the impact of overlap and contradiction.

3.4.1 Choice of Dataset

The transformer model predicts the next token of a given sequence[11]. While this sequence can take many forms from a paragraph of a book to a sequence of characters, the underlying working remains the same. On one hand, while the model can be trained on "War and Peace", a 560,000-587,000-word book [70] written by Leo Tolstoy, on the other hand, it can also be trained on a sequence of 6-10 characters to achieve the same objective: predict the next token in the sequence. Therefore, to work within the computational resources available, the latter was chosen for this study.

3.4.2 Design of Training Dataset - The "Songs"

The character sequences, in other words, "songs", were chosen to introduce varying levels of complexity and variability in the training data. The songs constructed are as follows:

1. **Normal:** "ABCDEFGHIJABCDEFGHIJABCDEFGHIJABCDEFGHIJAB" – This is a straightforward sequence that repeats in a predictable and linear fashion. It contains all characters from A to J assembled in an ascending order.

2. **Reverse:** "JIHGFEDCBAJIHGFEDCBAJIHGFEDCBAJIHGFEDCBAJI" – The same sequence as the first song but reversed, introducing a different pattern of transitions between notes.

3. **Daisy bell:** "CCGGAAGFFEEDDCGGFFEEDGGFFEEDCCGGAAGFFEEDDC" – A melodic pattern that mimics the structure of the classic tune "Daisy Bell", featuring repetitive and rhythmic sequences.

4. **10th Note Different:** "ABCDEFGHIEABCDEFGHIEABCDEFGHIEABCDEFG--HIEAB" – Similar to the normal song, but changes subtly at every 10th note, to test the model's attention to slight variations within patterns.

5. **Random Song:** "ABCDAFGHIJBCDEAGHIAFCDEFIHIABADEFGCIABCBEF" – This sequence is designed to create randomness to challenge the model with inconsistent training signals.

6. **5th Note Different:** "ABCDAFGHIAABCDAFGHIAABCDAFGHIAABCDAFG--HIAAB" – Similar to the normal song, but changes at every fifth note, to test the model's attention to more variations within patterns.

The song structures in our study were specifically chosen to test how effectively a model can learn from data with different levels of consistency and conflicts. How the model reacts to these variations is crucial for evaluating its ability to accurately predict and generalize across diverse datasets.

3.4.3 Data Extraction

The method involved converting strings of musical notes into a format suitable for our model. Each song was processed to create sequences represented numerically. For each song, a sequence of source notes was generated alongside a corresponding target sequence. This approach allowed the model to learn from the structured data by predicting the target sequences from the given source sequences. For a given length of 'source' and 'target', a manually created function cycles through all possible starting points of the song to capture all possible sub-sequences and their corresponding next notes. The output of this function is two arrays: one for the source sequences and one for the targets. Both are formatted as NumPy arrays for efficient handling during the training process.

3.4.4 Data Mixing and Shuffling

The method employed adjusts the training set by creating varied mixtures of the data based on a predetermined fraction. Initially, specifically designed function calculates the total number of sequences available from the prepared data. It then determines how many sequences should be allocated to two separate sets according to the specified fraction. A

random selection process is used to choose indices for these sets, ensuring each is unique and chosen without repetition. These indices are then shuffled to guarantee a random distribution of sequences, which is essential for preventing biases during model training. The shuffled sequences are then ready for use in training the predictive model. This approach ensures the training data is effectively mixed and shuffled, setting a strong foundation for robust model training. This method, however, was not employed in sequential learning as it did not involve data mixing.

3.5 Model Training Process

3.5.1 Training Setup for size, complexity, overlap, and contradiction

Thirty Transformer models were trained to assess the stability and variability in learning across different fractions. The training dataset consists of sequences derived from two specific songs for each case. These sequences were then mixed in varying proportions to simulate different training conditions. The mixing fractions range from 0% to 100%, in increments of 10%. The training procedure was implemented for each fraction of the dataset.

3.5.2 Training Setup for sequential learning

In this approach, thirty (for section 4.5) and fifty (for section 4.6) Transformer models were trained using a dataset that includes source and target sequences. The training procedure involves each model being trained sequentially, where the output of one model is used as the training data for the subsequent model. The below-mentioned training procedure was extended to a sequence of models, enabling a cascading learning effect. Each model builds upon the knowledge of the previous one, simulating a continuation of the initially-learned character sequence.

3.5.3 Training Procedure

The general procedure for training one model is as follows:

1. **Model Initialization:** A new Transformer model[10] was instantiated with the specified parameters and set to training mode.

2. **Loss and Optimizer Setup:** The cross-entropy loss[71] was used as the criterion, and the Adam optimizer[72] was initialized with the set learning rate.

3. **Data Preparation:** The input and target sequences were shuffled and split according to the current fraction to simulate different training scenarios.

4. **Epoch Training:**

- For each epoch, gradients were reset and a forward pass was performed to compute predictions from the input sequences.
- The loss was calculated by comparing the predicted output to the target sequences.
- Back propagation[73] was performed to update the model weights.

5. **Model Saving:** After completing the training for each fraction, the model's state was saved.

To ensure robust learning during test cases for overlap and contradiction, the input and output lists were re-shuffled[74] after each epoch. For sequential learning, neither fractions nor shuffling was used at any stage of training.

3.6 Model Testing Process

3.6.1 Testing Setup

Models were stored in a structured directory format and were trained on different fractions of the dataset. Each model was then tested on the same song sequences used during training.

3.6.2 Testing Procedure for size, complexity, overlap, and contradiction

The testing procedure involved several steps:

1. **Model and Data Loading:**
   - For each model, trained on a specific fraction of the dataset, the saved model state was loaded. For testing during sequential learning, this step was omitted as it did not involve varying fractions.
   - The musical sequences were processed to generate the appropriate source and target tensor pairs for a single song, which were then used for testing.
2. **Evaluation:**
   - Each model was evaluated on the test data by performing the following steps:
     (a) The model was set to evaluation mode using `model.eval()`[75].
     (b) The model's predictions were computed without tracking gradients using `torch.no_grad()`.
     (c) Predictions were compared to the target tensors, and the number of correct predictions was recorded.

3.6.3 Testing Procedure for sequential learning

After training, the process for sequential learning begins by predicting a specific sequence and then iteratively using these predictions as the new source. The model's performance is assessed by generating predictions for the provided source sequences. Predictions are compared to the actual target sequences, and accuracy is calculated as the percentage of correct predictions through element-wise comparison. To create source sequences for subsequent models, predictions from the trained model are used. This process is repeated for 50 models. Each model is trained using the source which is derived from the previous model's predictions. The accuracy of each model is recorded to evaluate performance over iterations.

3.7 Calculation of Statistical Metrics

In our study, we evaluate Transformer models by calculating the mean accuracy[76]. In sequential learning, the mean accuracy for both songs together was calculated for each model. In the remaining test cases, mean accuracy was calculated for each song and fraction by averaging the accuracy from multiple models. Each calculated mean accuracy is then expressed as a percentage. We also computed the standard error[77] of the mean to quantify the variation in model accuracy across different training iterations.The standard error is used to calculate the 95% confidence intervals around the mean accuracy, which contains the true population mean. These intervals are defined as ±1.96 times the standard error from the mean. To ensure consistency with the mean accuracy metric, these standard errors are expressed in percentage terms.

3.7.1 Decision Tree for insights into overlap and contradiction

The decision tree[78] classifier was trained using sequences derived from either a single song or a combination of multiple songs, consistent with the training approach used for the Transformer models. The trained classifier was then evaluated on sequences from different songs to assess its predictive accuracy across varying contexts. However, it is important to note that the decision tree classifier's assessment does not give maximum accuracy. This is due to the classifier's inability to effectively utilize all available samples when trained on mixed data fractions. In contrast, the transformers see all the training data. This strategy of using decision tree provides insights into highest mean accuracy achievable depending on the degree of overlap and contradiction. We use mean accuracy as a proxy to show how overlap and contradiction differ with transformers.

4 Results

In this study, we set our aim to investigate,

- The impact of Sequential learning: The consequences of AI-generated (potentially erroneous) output subsequently entering their own training dataset and,
- Model behaviour around Contradictory and Overlapping Information: Robustness and learning ability of LLMs when faced with contradictory and overlapping information along with model sizes and data complexities.

The results section that follows will introduce our findings under the aforementioned aims serving as the broad categories. These two categories will be further subdivided to discuss in more detail the results of our experiments. The first category discusses results obtained for transformer models' learning behaviour in the face of contradictory and overlapping information. Next, sequential learning set out to understand the effects of model-generated content entering its own training data - we discuss the results based on two possible ways this could happen.

4.1 Model Size

In this experiment we wanted to see if the size of the model plays a role in the learning ability of the model. The size of the model is varied by varying the embedding size, hidden dimensions, and the number of layers it contains.

Figure 1 illustrates the epochs required to reach 80% accuracy for six different songs using three transformer models with distinct parameters. We used 80% as the threshold because songs 3 and 5 can able to reach only 85% because of its complex nature which we covered in the next section. The models differ mainly in the size of their embedding, number of hidden dimensions and the number of layers, with Model 1 being the smaller one and Model 3 is the larger one.

Model 1, the smaller model, demonstrates that it can quickly reach 80% accuracy for most songs, with the notable exception of Song 3 and Song 5, where it requires a significantly higher number of epochs. This suggests that while Model 1 can handle simpler patterns but it struggles with more complex structures that might be present in Song 3 and Song 5.

Model 2, the medium sized model, demonstrates that it can quickly reach 80% accuracy for most songs, with the notable exception of Song 3 and Song 5 where it requires a significantly higher number of epochs. This suggests that while Model 2 can handle simpler patterns efficiently than Model 1 but it finds it difficult with more complex structures that might be present in Song 3 and Song 5.

Model 3, the larger and presumably more capable model due to its additional layers, more hidden dimensions and larger embedding size, also exhibits variable performance

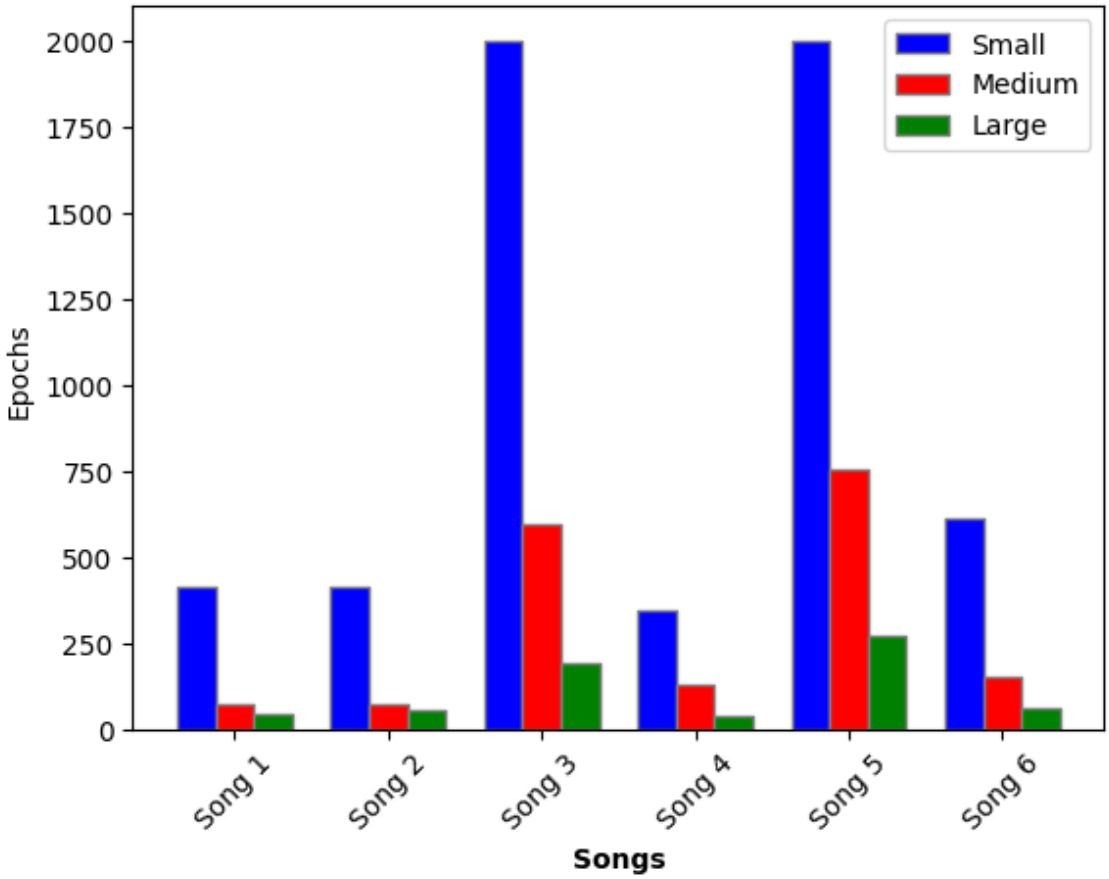

Figure 1: Model size and epochs to reach 80% accuracy to learn different songs, categorized as small (emsize $= 4$, d_hid $= 4$, nlayers $= 1$), medium (emsize $= 8$, d_hid $= 8$, nlayers $= 2$), and large (emsize $= 16$, d_hid $= 16$, nlayers $= 2$).

across the songs. It handles Song 3 more efficiently than Model 1 and Model 2, which aligns with the understanding that a larger model should theoretically learn complex patterns better[10].

4.1.1 Model size and its effects on different fraction of songs

The impact of model size on the performance of Transformer models trained on mixed data from two distinct songs is evaluated. The experiments were structured to determine how variations in model parameters—specifically, embedding size, hidden dimensions, and the number of layers—affected the model's ability to accurately predict sequences based on the training data. Here we categorized the models into small (emsize $= 4$, d_hid = 4, nlayers $= 1$), medium (emsize $= 8$, d_hid = 8, nlayers $= 2$), and large model (emsize $= 16$, d_hid = 16, nlayers $= 2$).

Note: It is important to note that the line with the shadow represents the average accuracy with respect to transformers, and the line with the error bar represents average accuracy with respect to the decision tree. X-axis at the bottom and the top represents the fraction of songs taken for that specific experiment. These figure shows the average

accuracy (%) of Transformer models (line with shadow) and Decision Tree models (line with error bar) on different fractions of mixed training data from two distinct songs. The shadow and the error bars represent the 95% confidence interval around the mean accuracy, calculated as approximately ±1.96 times the standard error from the mean. This holds true for section 4.1.1, 4.2.1, 4.3.

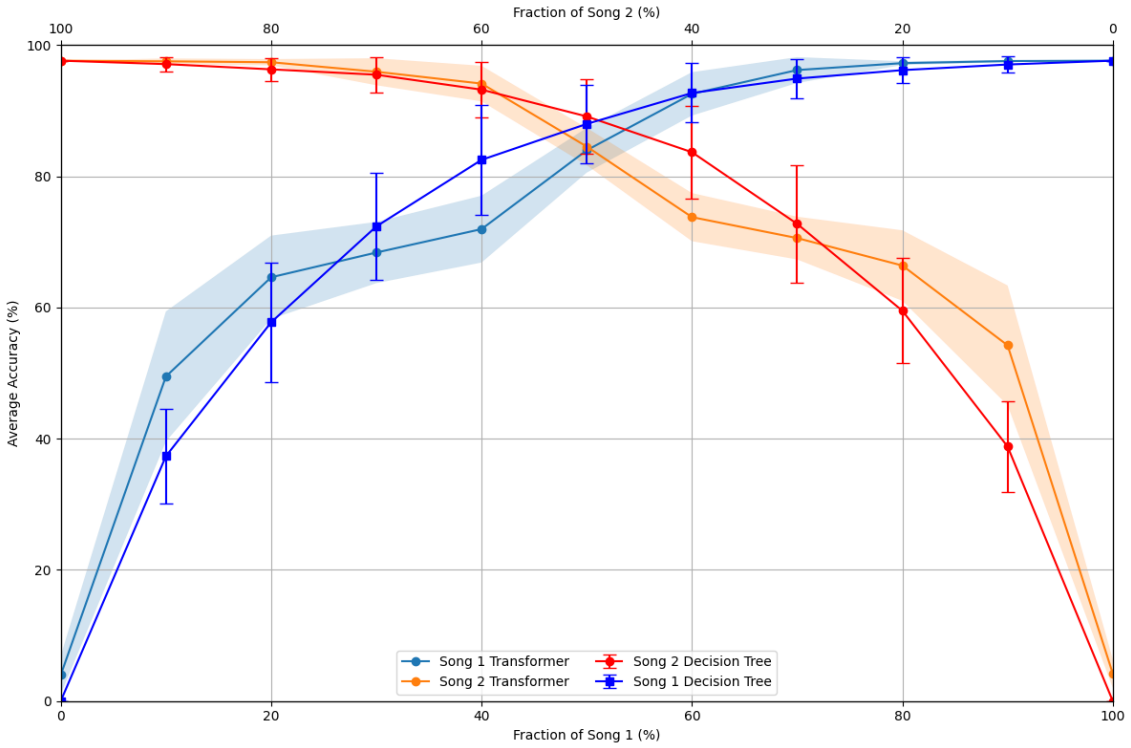

Figure 2: Average accuracy of the small model (embedding size = 4, hidden dimensions = 4, layers = 1) for Transformer (shadowed line) and Decision Tree (error-barred line) models across varying training data fractions from two songs. Shadows and error bars indicate the 95% confidence interval, ±1.96 standard errors from the mean.

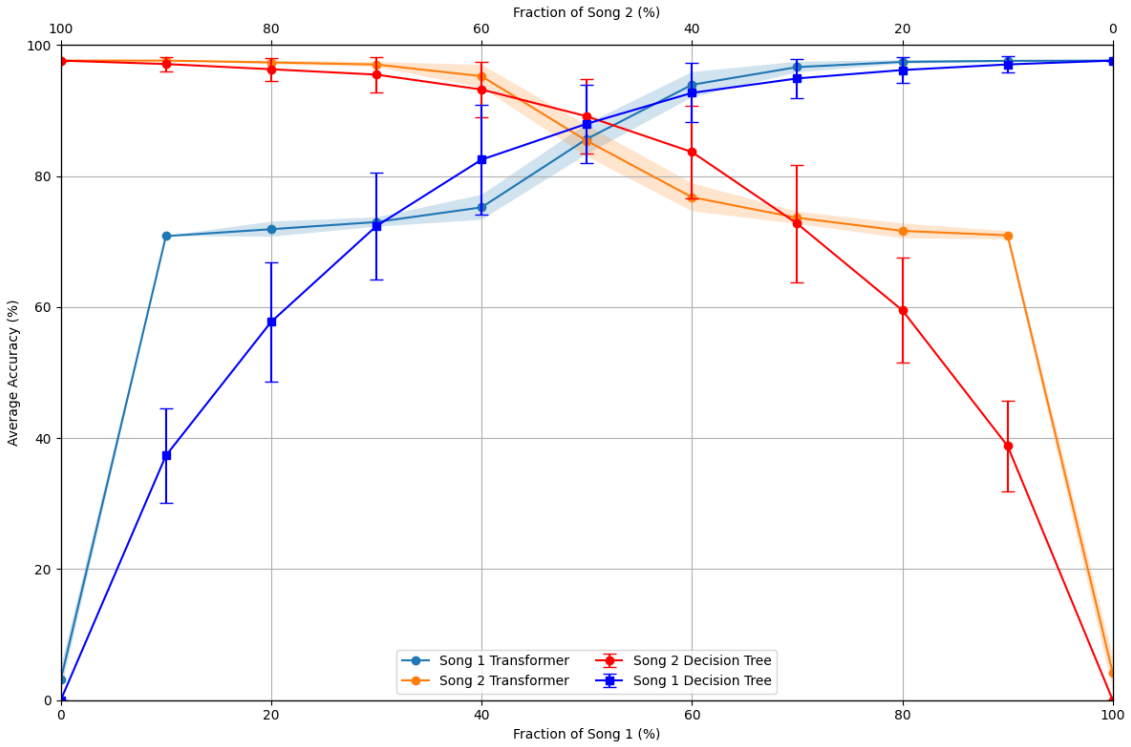

Figure 3: Average accuracy of the medium model (embedding size = 8, hidden dimensions = 8, layers = 2) for Transformer (shadowed line) and Decision Tree (error-barred line) models across varying training data fractions from two songs. Shadows and error bars indicate the 95% confidence interval, ±1.96 standard errors from the mean.

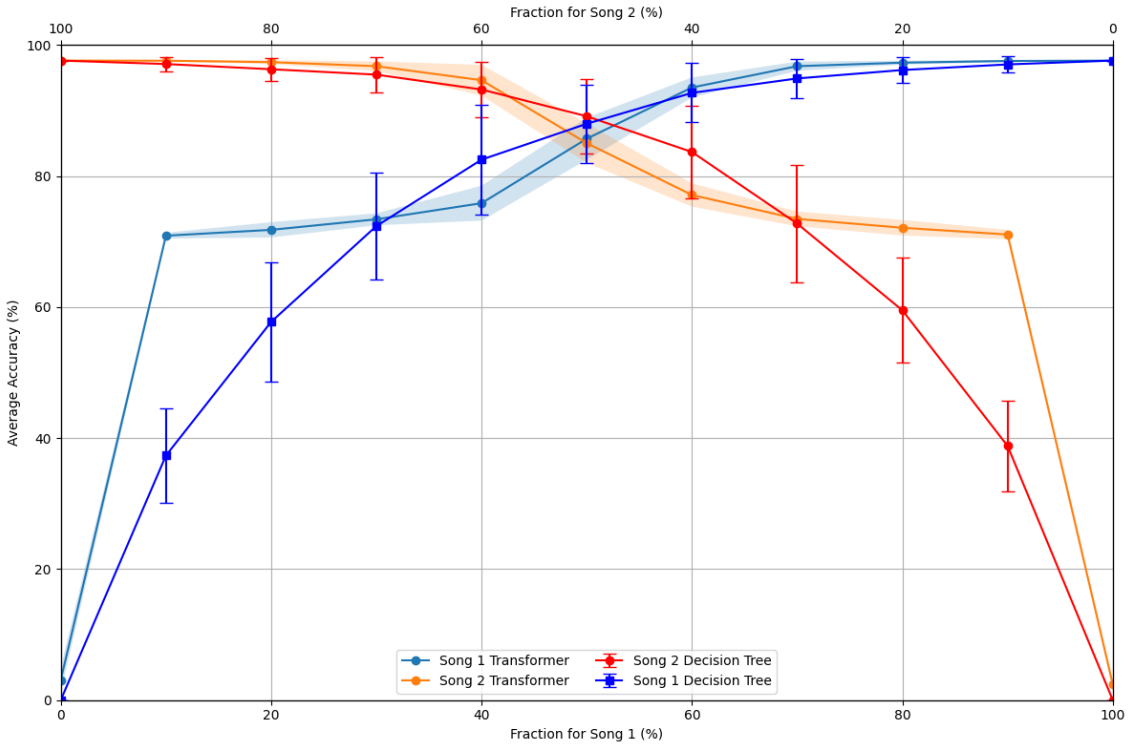

Figure 4: Average accuracy of the large model (embedding size = 16, hidden dimensions = 16, layers = 4) for Transformer (shadowed line) and Decision Tree (error-barred line) models across varying training data fractions from two songs. Shadows and error bars indicate the 95% confidence interval, ±1.96 standard errors from the mean.

Before diving into the results, it's important to outline our expectations based on the model size variations. We hypothesized that increasing the model size would enhance the model's ability to learn and generalize from the training data. Specifically, we anticipated that larger models, with more parameters and higher capacity, would perform better in accurately predicting sequences, even with skewed training distributions. Smaller models, with limited capacity, were expected to struggle more, especially when trained with imbalanced data. As the model size increases, we expected a corresponding increase in accuracy, particularly evident when training data is evenly split between the two songs.

The performance data in figures 2, 3, 4 indicates that increasing the model size leads to improvements in accuracy[68]. For instance, when the model was trained with a significantly skewed distribution (90% from Song 2 and only 10% from Song 1), there was a notable increase in accuracy on Song 1, demonstrating the enhanced generalization capabilities of the larger model[10]. Balanced training data (50% from each song) further validated the model's improved learning and predictive capacity, with accuracies reaching approximately 85% for Song 1 and 86% for Song 2 in figure 3 from 83% and 84% as in figure 2.

For example, when the model was trained mostly with data from Song 2 (90%) and less from Song 1 (10%), it showed a noticeable improvement in accuracy for Song 1, highlighting its better prediction ability even through it was trained with only 10% from Song 1 . When the training data was evenly split between the two songs (50% from each), the model's learning and prediction skills were confirmed to improve, with accuracy rates reaching about 85% for Song 1 and 86% for Song 2, as shown in figure 3, compared to 83% and 84% previously in figure 2.

After seeing significant improvements by increasing model parameters, we tried making the models even bigger. However, this additional increase did not improve performance, as shown in figure 4. Accuracy levels stopped improving, suggesting that making the model bigger beyond a certain point no longer helps. This finding indicates that there is a limit to how much increasing model size can improve performance[68], showing that there is an optimal size for models beyond which making them larger doesn't make them more efficient or effective.

4.2 Song Complexity

In this experiment we wanted to see if the complexity of the data(songs in this case) plays a role in the learning ability of the model. The complexity of a song can be assessed by how challenging it is for the model to learn the song's patterns. This is measured by the duration of training needed for the model to achieve a certain level of accuracy and the ultimate performance it can attain within a given number of training epochs. Songs can range from simple, with easily discernible patterns, to complex, with intricate sequences that challenge the model's learning capacity.

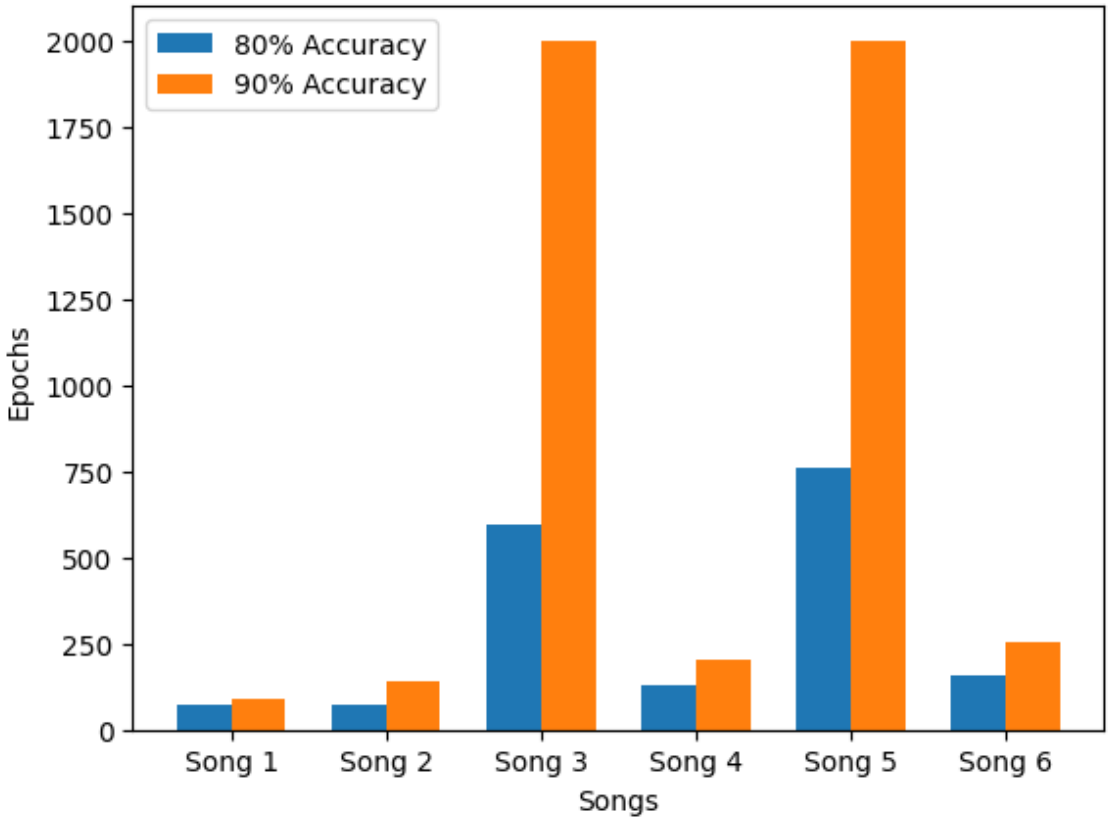

Figure 5: Epochs required by songs to reach 80% and 90% accuracy

Figure 5 represents the number of epochs required for a model to reach two specified accuracy thresholds—80% and 90%—for six different songs. Even though songs 3 and 5 do not reach 90% accuracy we used it as a threshold to have the comparison among other songs

For Songs 1, 2, 4, and 6, the chart shows relatively low bars for both accuracy levels, indicating that the model reaches both 80% and 90% accuracy quickly. This suggests that these songs are less complex, or the patterns within them are more easily learned by the model.

In contrast, Songs 3 and 5 require more epochs to reach the accuracy thresholds, particularly for the 95% accuracy, which is represented by a taller bar. While reaching 80% accuracy was within the capabilities of the model in a reasonable number of epochs, achieving 90% accuracy was more challenging. This points to higher complexity in the patterns and structures within these songs.

Songs 1, 2, 4, and 6 could be considered less complex or more predictable in their structure, making them easier for the model to learn. In contrast, Songs 3 and 5 present greater challenges, necessitating a larger number of training epochs to achieve high accuracy. This indicates a higher level of complexity in their musical patterns.

4.2.1 Song Complexity and its effects on different fraction of songs

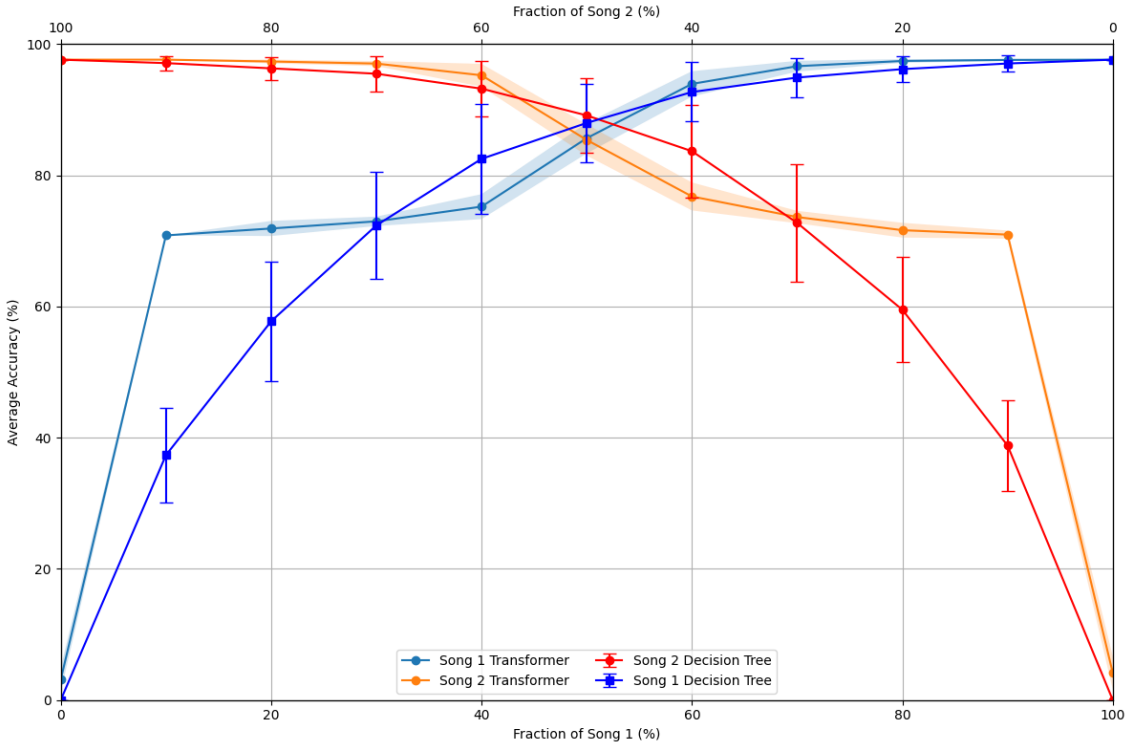

Figure 6: Average accuracy of models trained on straightforward sequences with clear, consistent patterns (easy songs) is displayed for Transformer (shadowed line) and Decision Tree (error-barred line) across varying training data fractions. Shadows and error bars signify the 95% confidence interval, calculated as ±1.96 standard errors from the mean.

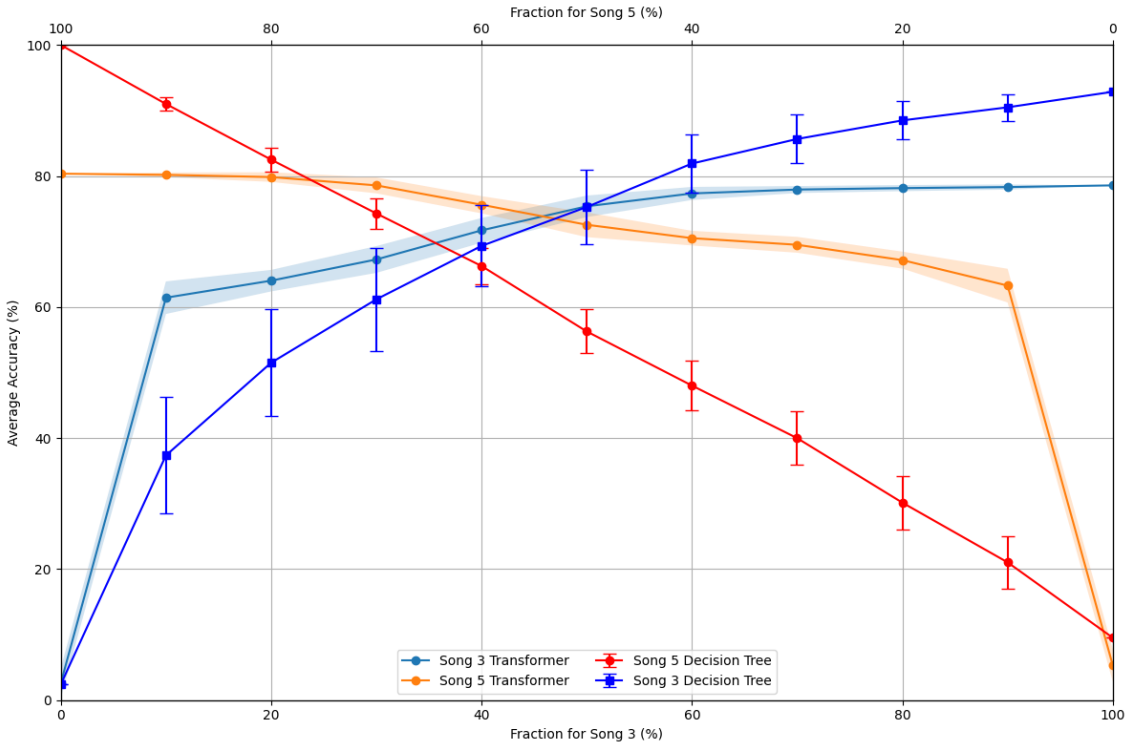

Figure 7: Average accuracy of models trained on complex songs with intricate patterns and variations (hard songs) is displayed for Transformer (shadowed line) and Decision Tree (error-barred line) across varying training data fractions. Shadows and error bars signify the 95% confidence interval, calculated as ±1.96 standard errors from the mean.

Here we hypothesized that when models are trained on straightforward musical sequences with clear, consistent patterns, they would perform exceptionally well, achieving high accuracy rates. Conversely, we expected that training on more complex songs with intricate patterns and variations would pose significant challenges, potentially reducing the models' accuracy. When trained on mixed data, we anticipated a slight decrease in performance due to the increased difficulty of generalizing across different patterns.

When trained on straightforward musical sequences, the Transformer models demonstrated exceptional learning capabilities. Individual song training showed models reaching near-maximal performance, with accuracy rates up to 98% when trained exclusively on one song as in figure 6. This indicates robust learning from consistent, patterned data[10]. On the other hand, mixed song training resulted in a slight decrease in accuracy to 85% for the normal song and 86% for the reversed version when models were trained with an equal mix of both songs. Although these are high performance metrics, the slight drop in accuracy suggests minor challenges in generalizing across two closely related yet distinct patterns.

The evaluation then extended to more complex songs, which presented a higher degree of learning difficulty. Individual song training showed that the models only achieved a maximum of 80% accuracy, significantly lower than their performance on simpler songs as shown in figure 7. This outcome illustrates the challenges posed by complex data struc-

tures, even when the models are exclusively focused on one type of sequence[79]. Mixed song training, when trained on a balanced mixture of both complex songs, saw further declines in accuracy to 76% for "Daisy Bells" and 72% for the song with random note variations. This reduction in performance indicates that complexity in training data can cause learning difficulties, impacting the models' ability to generalize effectively across diverse patterns.

4.3 Overlap and Contradiction

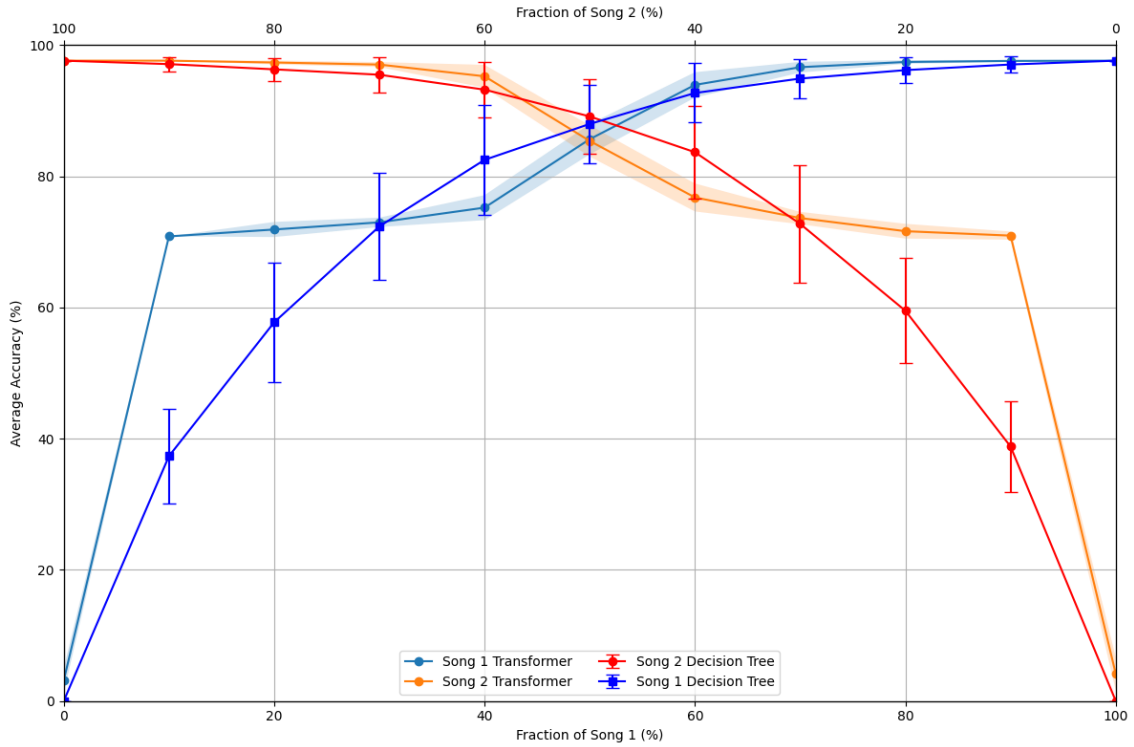

Figure 8: Average accuracy of models trained on a normal song and its reversed counterpart, which have no overlap or contradiction, is displayed for Transformer (shadowed line) and Decision Tree (error-barred line) across varying training data fractions. Shadows and error bars signify the 95% confidence interval, calculated as ±1.96 standard errors from the mean.

Before examining the results, we hypothesized that overlap and contradiction within the dataset would significantly influence the model's learning ability. We expected that models trained on non-overlapping datasets, such as a normal song and its reversed counterpart, would struggle to generalize, showing low accuracy, but learning both songs when trained with equal mixture. We anticipated that models would perform better with datasets containing minor contradictions due to the presence of some overlapping sequences. For datasets with frequent contradictions, we expected even greater challenges in learning and generalization.

In this experiment we wanted to see if the overlap and contradiction in the dataset plays a role in the learning ability of the model. Figure 8 shows models trained on a nor-

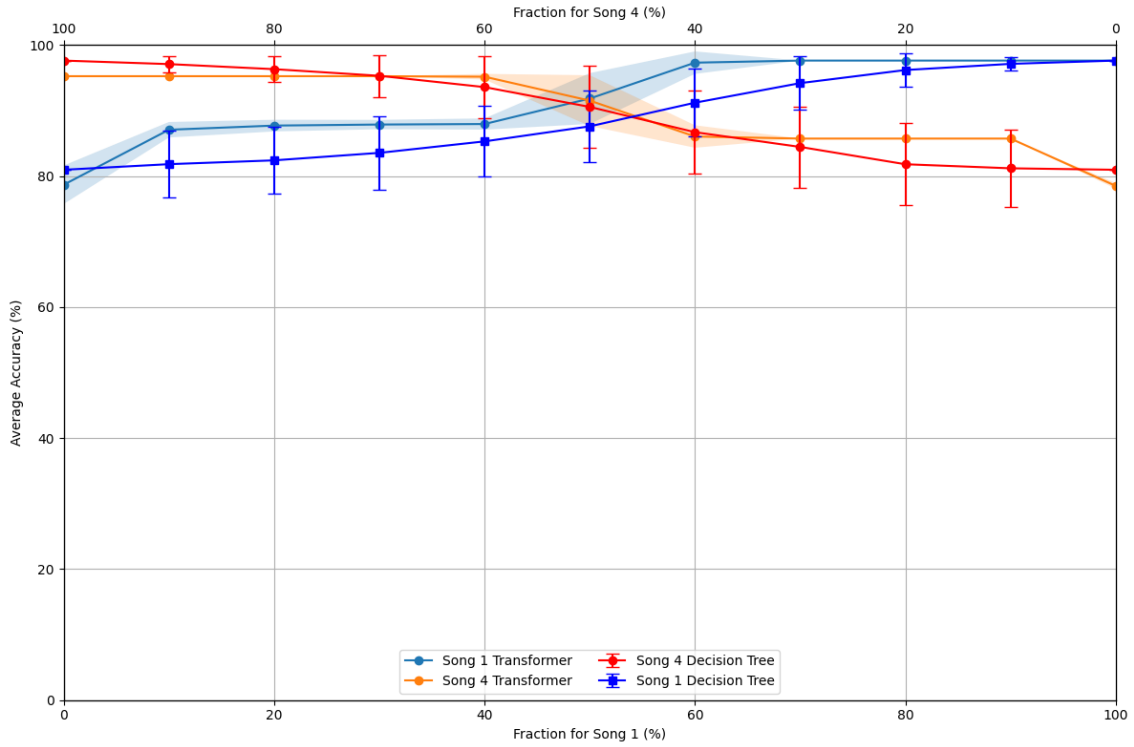

Figure 9: Average accuracy of models trained on datasets with overlap and slight contradiction, where every tenth note differs, is displayed for Transformer (shadowed line) and Decision Tree (error-barred line) across varying training data fractions. Shadows and error bars signify the 95% confidence interval, calculated as ±1.96 standard errors from the mean.

mal song and its reversed counterpart demonstrated significant challenges in generalizing across these non-overlapping datasets . When trained exclusively on one song, the models could not predict the sequences of the other, with accuracy close to zero. This illustrates the difficulty Transformers face when there is a complete lack of overlapping features between the training and test datasets. However, accuracy improved when the models were trained on a 50% mix of both songs, achieving up to 85% accuracy, still its less than the mean accuracy of the decision tree(91%).

In figure 9 where every tenth note differs, the presence of minor contradictions within an overall overlapping structure allowed for better model performance. Exclusive training on the modified song achieved a reasonable 78% accuracy on the original song, indicating effective learning from overlapping sequences despite the contradictions[79]. The best results were obtained when training involved an equal mix of both the original and its slightly modified version, with accuracy reaching 91%, which is almost equal to the mean accuracy of the decision tree. This outcome demonstrates the model's capacity to effectively manage minor contradictions by leveraging underlying similarities.

Training on a song where every fifth note was altered posed a greater challenge, reducing model accuracy to 63% when tested on the standard song as shown in figure 10, but it was greater than the mean accuracy from the decision tree results. This indicates that

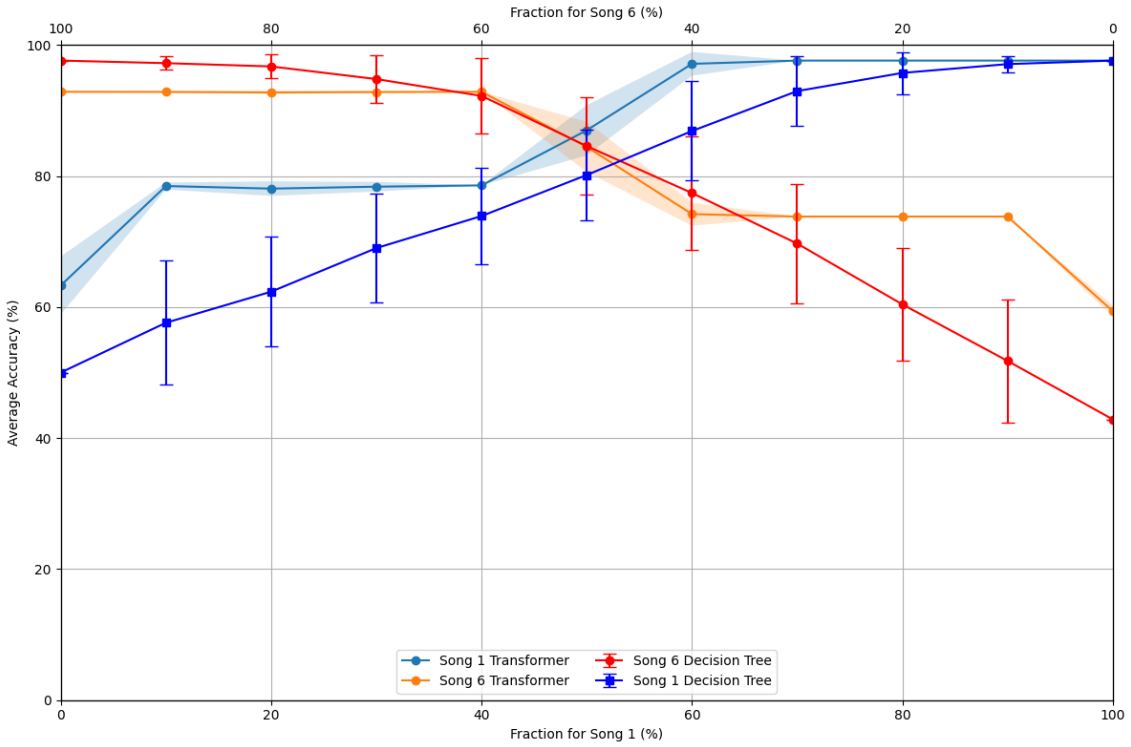

Figure 10: Average accuracy of models trained on datasets with overlap and more contradiction, where every fifth note was altered, is displayed for Transformer (shadowed line) and Decision Tree (error-barred line) across varying training data fractions. Shadows and error bars signify the 95% confidence interval, calculated as ±1.96 standard errors from the mean.

more frequent alterations intensify learning difficulties and hamper generalization [79]. Despite these challenges, balancing the training with equal parts of the standard and more heavily altered song improved performance, with 87% and 84% accuracy for song 1 and song 5 respectively, which is higher than the mean accuracy of the decision tree, but it was slightly less compared to the results of minor contradiction experiment.

4.4 Sequential Learning - Overview

As explained in the methodology section, the sequential learning process was designed to investigate the consequence of a model's (possibly erroneous) output entering its own training dataset. We can imagine that these errors can be caused not only by a model's own output, but possibly other similar AI models as well. therefore, in practice, one could imagine more than one possible approach to designing the experiment. This study focuses on two such variations:

- Learning from the complete song - 'seeing' the complete song predicted by the previous model and making its prediction for the next step of the song
- Building the song using only a snippet - 'seeing' a snippet of the song generated by the previous model, and building the rest of the song by itself

The variations along with the song-pairs considered are shown in table below.

Experiment variation	Song-Pair Used
Contain no overlap or contradiction	Normal and Reverse
Overlap with moderate level of contradiction	Normal and every 10th note different
Overlap with higher level of contradiction	Normal and every 5th note different

Table 1: Song-pairs used as datasets in sequential learning

In both experiments, the song-pairs were concatenated into a single matrix for their source and target. The models were trained on sliding windows of 4 notes at a time. The difference between these experiments are explained below.

4.5 Learning from the complete song

As we know, the transformer predicts the next token of a sequence. Consider the "normal" song, "ABCDEFGHIJABCDEFGHIJABCDEFGHIJABCDEFGHIJ". Once encoded, it will take the form before:

Song	A	B	C	D	E	F	G	H	I	J	A	B	C	D	E ...
Encoded	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4 ...

Table 2: "Normal" Song - First 15 Notes of the Song with Encoded Values

Transformed into source and target, the song will take the form:

Song	**Source**		**Target**	
	Characters	**Encoded**	**Characters**	**Encoded**
Song 1	A B C D	0 1 2 3	B C D E	1 2 3 4
	B C D E	1 2 3 4	C D E F	2 3 4 5
	C D E F	2 3 4 5	D E F G	3 4 5 6
	D E F G	3 4 5 6	E F G H	4 5 6 7
	. . . .	. . . .	. . . .	. . . .
Song 2	J I H G	9 8 7 6	I H G F	8 7 6 5
	I H G F	8 7 6 5	H G F E	7 6 5 4
	H G F E	7 6 5 4	G F E D	6 5 4 3
	G F E D	6 5 4 3	F E D C	5 4 3 2
	. . . .	. . . .	. . . .	. . . .

Table 3: An Example of a source and target matrix after mapping characters of two songs to encoded values. The transformer model is required to predict the two target songs based on the provided source songs

As table 3 indicates, for the sliding window of "ABCD", the transformer learns to predict the target sequence, "BCDE". Each sliding window is designed such that it starts

from the next position of the original song. In effect, this design will enable the model to learn to predict the song note-by-note. Continuing this process for the entire source matrix, we expect the model's output to result in the target matrix.

The initial model was trained on the original source and target - the original song, and its output was used as the source to train the next model. Each subsequent model was thus sequentially trained on the predicted output of its predecessor. This training setup lets each model predict one step further than the other. It results in a continuation of the "song," as if the models were each singing the next note of the song, one after the other. Since the source is two songs concatenated together, our interest lies in finding out how well the models learn the song-pair, how far the models continue 'singing' the songs, and whether they lose accuracy or maintain it down the line.

4.5.1 Song-pair with no overlap or contradiction

The first song pair - Normal and Reverse, share the least similarities between them. They neither overlap nor contradict with each other. As figure 11 illustrates, most models' learning reached a plateau between approximately 500-1000 epochs. When tested against the expected target, the initial model managed to score an accuracy of 85.41% when learning the two songs together. The second, third, and fourth models scored an accuracy of 71.72%, 61.6%, 67.85%, respectively.

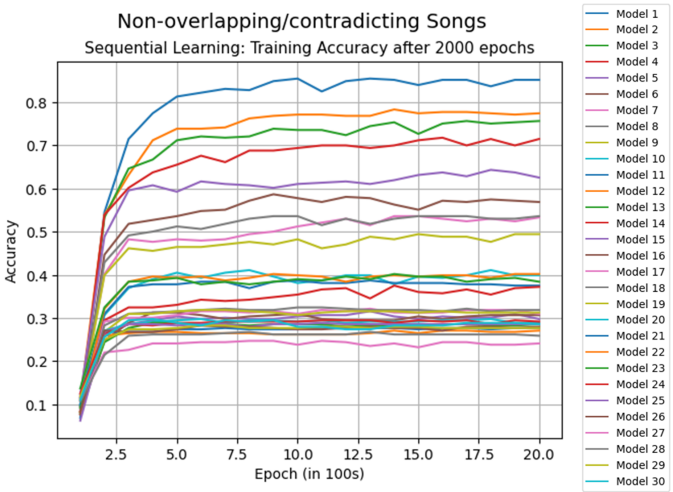

Figure 11: Individual Accuracy for Non-Contradicting Songs: This figure displays the individual model training accuracy of sequential learning for two non-contradicting songs (normal song and reverse song (section 3.4.2). Each model sequentially learns from the output of the previous model and the overall accuracy for both songs together can be seen by the colored lines. Accuracy of model 1 is shown by the top-most blue line, followed by the accuracy of model 2 which learned from model 1, followed by the accuracy of model 3 which learned from model 2,and likewise.

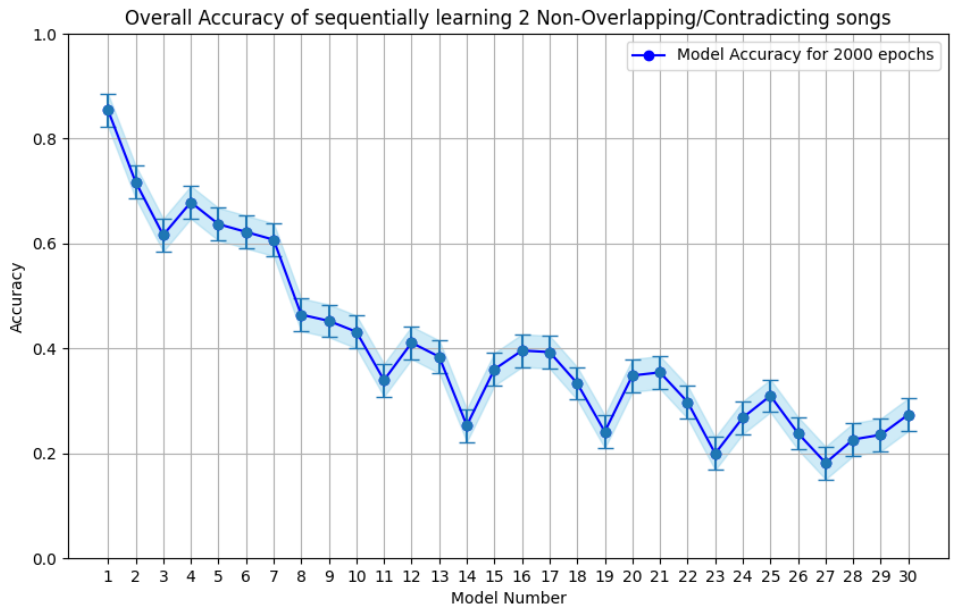

Figure 12: Overall Accuracy for Non-Contradicting Songs: This figure depicts the overall accuracy of sequential learning for two non-contradicting songs (normal song and reverse song (section 3.4.2). In contrast to figure 11, this offers an overall outlook on the effect of models sequentially learning from their predecessor's output. The shadow and the error bars represent the 95% confidence interval around the mean accuracy, calculated as approximately ±1.96 times the standard error from the mean.

Figure 12 depicts the impact of each model learning from the output of the previous model. While there is some fluctuation of accuracy, the deterioration of prediction accuracy can be seen clearly.

4.5.2 Song-pair with overlap and moderate level of contradiction

Comparatively, learning two songs with moderate levels of overlap and contradiction (Normal and 10^{th} note different song) appeared easier for models. Figure 13 shows that all models reached their maximum accuracy within 250-500 epochs. The initial model reached an accuracy of 91.66%, the second, 88.1%, and the three subsequent models scored 83.3% each.

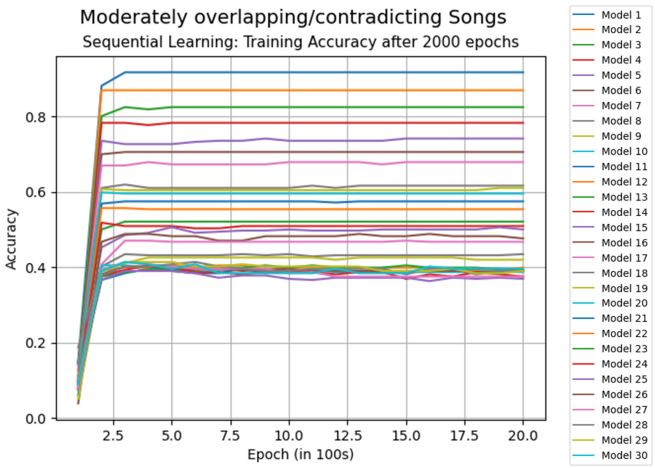

Figure 13: Individual Accuracy for Overlapping and Moderately Contradicting Songs: This figure displays the individual model training accuracy of sequential learning for two overlapping songs with moderate levels of contradiction (normal song and 10th note different song (section 3.4.2). Each model sequentially learns from the output of the previous model and the overall accuracy for both songs together can be seen by the colored lines.

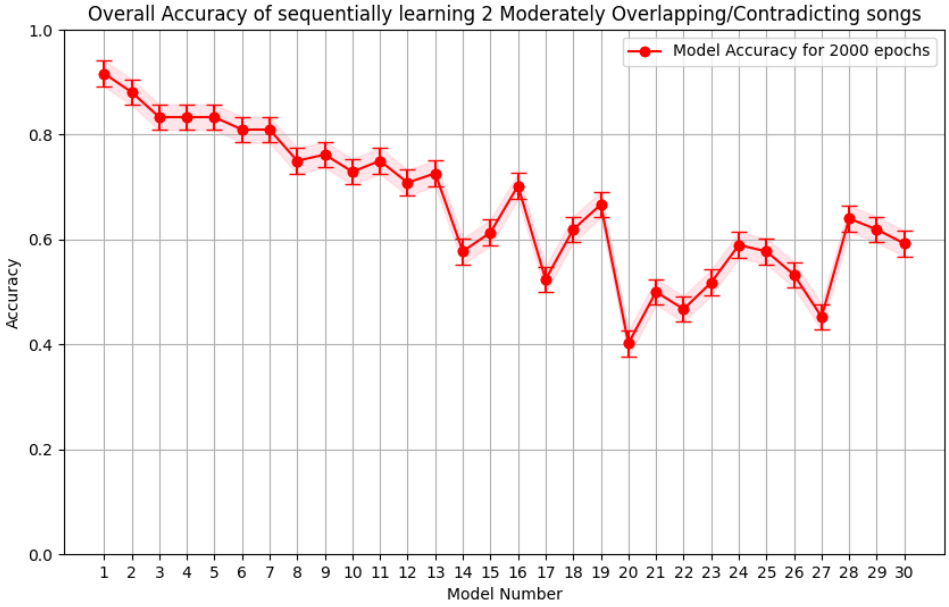

Figure 14: Overall Accuracy for Overlapping and Moderately Contradicting Songs: This figure depicts the overall accuracy of sequential learning for two overlapping and moderately contradicting songs (normal song and 10th note different song (section 3.4.2). The shadow and the error bars represent the 95% confidence interval around the mean accuracy, calculated as approximately ±1.96 times the standard error from the mean.

The overall accuracy decline of this song-pair seen in figure 14 was slower than the one with no contradictions or overlaps, but displayed more fluctuations after model 13.

4.5.3 Song-pair with overlap and higher level of contradiction

The third experiment conducted with the normal and a song where every 5th note was different contained the most contradictions out of all three song pairs. Over time, it proved harder for the models to learn and predict as accurately as the previous song pair with fewer contradictions. The initial model reached a maximum accuracy of 85.71% while the subsequent models' accuracy resided between 70-80%.

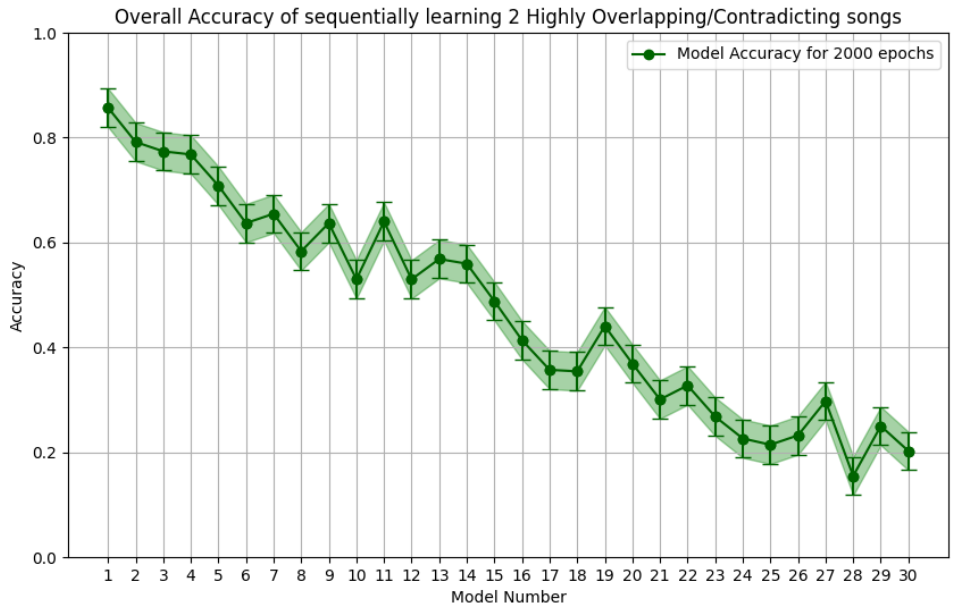

Figure 15: Overall Accuracy for Overlapping and Highly Contradicting Songs: This figure depicts the overall accuracy of sequential learning for two overlapping and highly contradicting songs (normal song and 5^{th} note different song (section 3.4.2). The shadow and the error bars represent the 95% confidence interval around the mean accuracy, calculated as approximately ±1.96 times the standard error from the mean.

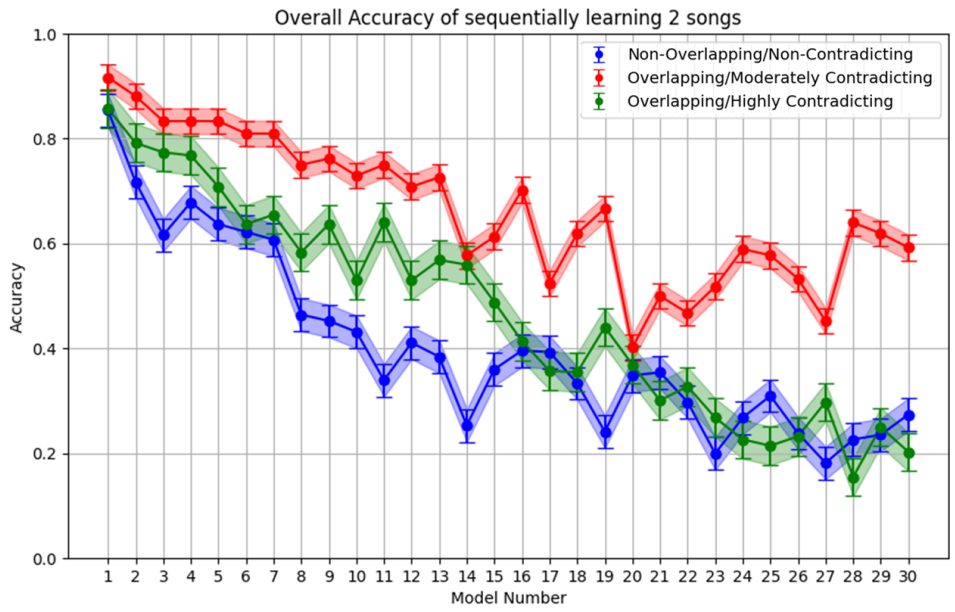

Figure 16: Overall Accuracy of Sequential Learning for All Song-Pairs in Comparison: This figure depicts the overall accuracy of sequential learning the different song-pairs mentioned in figures 12, 14, and 15). The shadow and the error bars represent the 95% confidence interval around the mean accuracy, calculated as approximately ±1.96 times the standard error from the mean.

Compared to the song-pair with fewer contradictions, the prediction accuracy for this experiment displayed a steeper decline. The results in figure15 indicate that models find it more difficult to learn sequences with a higher number of contradictions between them.

Figure 16 offers a comparative view of prediction accuracy deterioration among the three song-pairs. It can be seen that the prediction accuracy is sustained longest when the songs overlap with fewer contradictions. As contradictions increase, the predictions deteriorate faster and steeper. It is also noteworthy to see that the decline is fastest when the songs have neither overlaps nor contradictions in common. It hints at the notion that the larger the difference between two concepts/sequences, the harder they are to learn together, even with more time.

4.6 Building the song using only a snippet

In this experiment, we ask the following questions in that order, "How well can a model predict the entire sequence by learning from smaller sections of its own previous output? Once the model builds the complete song and if that song is used as training data for the next model, how accurately can the next model recreate the original song?" As outcome, we expected to see a larger decline in prediction accuracy and increasingly inaccurate reproductions of the original song within the sequential learning. The reason for this can be understood from the design of the experiment explained below.

The original source and target was designed as explained in table 3. Similar to the previous experiment, we provided the entire source (complete matrix which includes all possible 4-character sequences for both songs) and target. However, when testing the model's predictions, instead of the entire source matrix, we provided the model the last sequence of each song.

For example, for the normal song, [0 1 2 3] represents the start of the song and [9 0 1 2] its end (see table 2). When the song begins with [0 1 2 3], [1 2 3 4],... and continues that way throughout the matrix, it ends with ..., [8 9 0 1], [9 0 1 2] so that it can loop back to [0 1 2 3] again. Thus, given [9 0 1 2] as the input, we expect the model to provide us the output, [0 1 2 3]. Essentially, we are prompting the model with the last "word" of the song so that the model starts "singing" the song from the beginning - If the song is "Happy birthday to you," we prompt the model with "you" for it to start singing the song from "Happy."

We start by training the first model using the original source and target created from the original song. Once trained, we prompt the same model with the last sequence of the original song (for example, [9 0 1 2]) so that it predicts the next sequence (ideally, [0 1 2 3]). Upon doing so, we provide that sequence which the model produced as its new input (assume, [0 1 1 3]) which may or may not be completely accurate. Regardless, based on that, the model predicts the next sequence (possibly, [1 1 3 4]), and the process continues for the entire length of the song. In this manner, the model receives its own previous output as the new input, and starts building the song, one sequence at a time - hence the name, "forward-prediction." The end result is an output of what the model "thinks" the song is. We then provide the constructed song as the new training input for the next model, against the original target. For all models in this training cascade, while the training source becomes the song constructed by the previous model, the training target remains the original. Moreover, the starting prompt is the last sequence of the constructed song so that each model starts predicting the song from its beginning.

While the above example uses one song for clarity, it should be noted that we train each model on song-pairs stated in table 1. The same process is extended so that the model trains on both songs, then constructs not one, but two songs. This training was sequentially conducted for 50 models in each experiment.

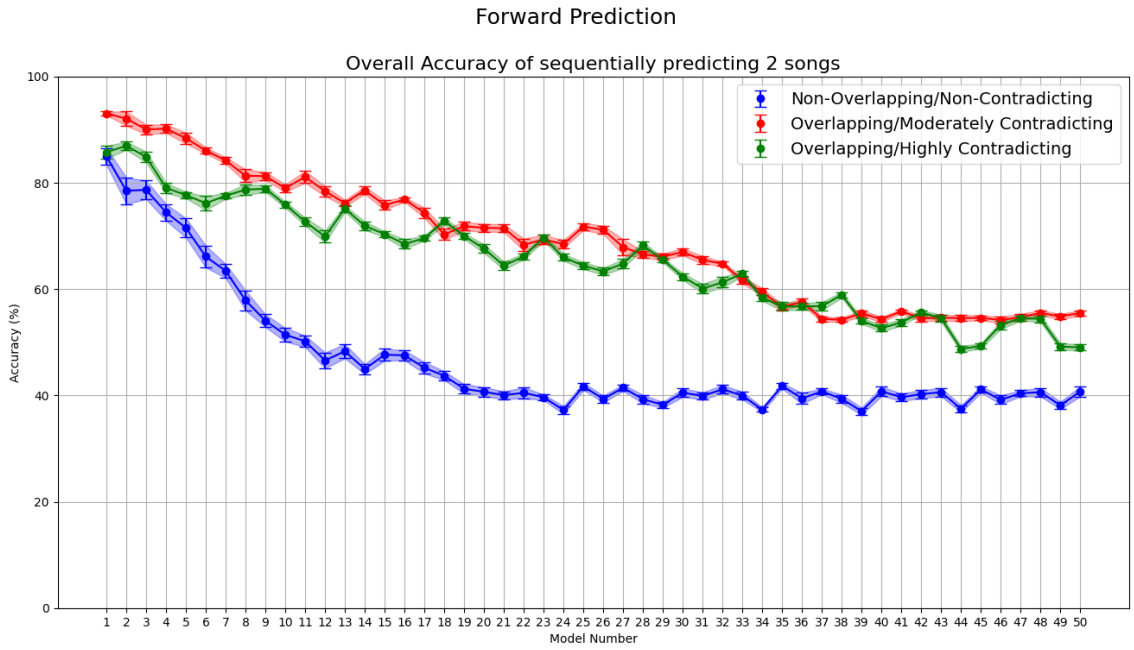

Figure 17: Forward Prediction Accuracy - All song-pairs in comparison: This figure depicts the overall accuracy of forward predicting using snippets of songs for different song-pairs, Non-contradicting (Normal and reverse song), overlapping and moderately contradicting (normal song and 10th note different song), overlapping and highly contradicting (normal song and 5th note different song). The shadow and the error bars represent the 95% confidence interval around the mean accuracy, calculated as approximately ±1.96 times the standard error from the mean.

4.6.1 Song-pair with no overlap or contradiction

Figure 17 illustrates in blue, the decline in overall accuracy for the non-overlapping/non-contradicting song-pair. The first model achieved an overall accuracy of 87.5% and the 50th model reached 42.5% overall. While there is a clear degradation of accuracy, it stabilizes around 40% after model 20.

4.6.2 Song-pair with overlap and moderate level of contradiction

The red plot line in figure 17 portrays how overall model prediction accuracy for overlapping/moderately contradicting song-pair is slower than that in the non-overlapping/non-contradicting song-pair. Since the two songs share a higher number of common notes (characters) throughout their sequences, the model does not need to exert as much effort to learn them compared to songs with no shared notes. The first three models achieved an overall accuracy of 93.04%, 92.06%, 90.03%, respectively. The last three models reached 55.52%, 54.84%, and 55.5%.

From model 35 onwards, the prediction accuracy stabilizes around 50-60%, which is higher than the 35-45% stabilization range in non-overlapping/non-contradictory songs. It raises the idea of whether the models predict with less inaccuracy when there is some

overlap between the song pair, despite some contradictions. However, we dissect this thought in more detail in the section 4.6.4, Hallucination.

4.6.3 Song-pair with overlap and higher level of contradiction

In this experiment, the first model scored an accuracy of 85.75%, followed by models 2 and 3 scoring 86.95% and 84.859% respectively.The 49^{th} and 50^{th} models reached accuracies of 49.12% and 49.03%. The decline in accuracy for this song-pair (shown in green in figure 17) is similar to that with moderate contradiction. There also appears to be no real point of stabilization for this song-pair, rather, a constant decline in accuracy. One might argue that it could take longer to reach such a point due to the increased level of contradiction within the song pair.

When compared together, the song-pair with no overlaps or contradictions displayed the fastest deterioration in prediction out of all three while also having the lowest accuracy for the initial model. The song-pairs with moderate and high contradiction showed similar patterns, despite the latter having a lower starting accuracy. More overlaps between songs tend to benefit models learning in a sequence. Conversely, the greater the difference between two songs, the more challenging it is for a model to 'construct' the song independently, even when provided with a prompt.

4.6.4 Model Hallucination

A particular phenomena is observed in the outputs of the later models in both experiments, in addition deterioration in their predictions. The models begin to predict so inaccurately that they enter a state of 'hallucination'. It makes false predictions with low accuracy yet high precision. In other words, although the generated sequences deviate from the true sequence, they are precise because the predicted characters in the sequence are closer to each other.

For example, consider the case of constructing the song from a snippet at a time. While the initial sequence for the first song in the moderately contradicting song-pair is [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ...], model 35 predicts [0, 3, 4, 0, 1, 7, 3, 4, 0, 1, 2, 3, 4, 0, 1, 7, 3, 4, 0 ...] repeatedly. Notice the repeating pattern of [0, 3, 4, 0, 1, 7, 3, 4]. Further towards the end of the line, model 49 predicts [0, 8, 4, 5, 1, 2, 8, 4, 5, 1, 2, 8, 4, 5, 1, 2, 8, 4, 5, 1, ...]. Notice how the model repetitively produces the sequence [8 4 5 1 2]. It was observed that while the model produces incorrect repetitive patterns, the patterns decrease in length. If the original song may have been "Happy Birthday To You", now the model perhaps sings "You day," repeatedly.

Song	Model Number	Predicted Song
With Overlap and Moderate Contradiction	Target	0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 ...
	46	0 3 4 0 1 7 3 4 0 1 7 3 4 0 1 7 3 4 0 1 ...
	47	0 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 ...
	48	0 8 4 0 1 2 8 4 0 1 2 8 4 0 1 2 8 4 0 1 ...
	49	0 8 4 5 1 2 8 4 5 1 2 8 4 5 1 2 8 4 5 1 ...
	50	5 3 4 0 1 7 3 4 0 1 7 3 4 0 1 7 3 4 0 1 ...
With Overlap and High Contradiction	Target	0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 ...
	46	0 3 0 5 6 7 2 8 0 1 2 3 0 5 6 7 2 8 0 1 ...
	47	6 3 0 5 6 7 8 3 0 5 2 3 0 5 6 7 8 3 0 5 ...
	48	0 3 0 5 2 7 8 0 0 5 2 3 0 5 2 7 8 0 0 5 ...
	49	0 7 4 5 6 7 8 9 0 5 6 7 4 5 6 7 8 9 0 5 ...
	50	0 7 8 5 6 7 8 9 0 1 6 7 8 5 6 7 8 9 0 1 ...

Table 4: This table shows the first 15 notes of the predicted model outputs in forward prediction. The 'target' row shows the original target the model should have predicted, if it were accurate. Both targets begin with the sequence, [0,1,2,3,4,5,...] since the starting song for each song pair is the normal song. The second song for overlap and moderate contradiction is the one where each 10^{th} note is different, and for overlap and high contradiction, it is the one where each 5^{th} note is different.

The later models also return some accuracy due to having some or part of the old words from the original song in their output. Thus, the accuracy does not drop to zero. Consequently, in the case of 4.6, accuracies of the later models stabilize within a certain range. The models begin to hallucinate, predicting the increasingly incorrect outputs as the correct sequence for the song. It was observed in all experiments across the two training paradigms, and the resemblance of output was similar to the song-pair outputs shown in table 4. The 15 tokens in the final songs constructed by models 46-50 in forward-prediction are shown in the table for comparison against the original song in 2.

4.6.5 Expectation versus Reality

We mentioned that in this section 4.6, we expected to see a larger decline in prediction accuracy and increasingly inaccurate reproductions of the original song. While there was a decline in accuracy, it was not so large that it reached zero. Instead, we observed that it stabilized for two out of the three song-pairs. However, the reproductions of the original song reached a level of distortion we expected. We noticed the models entering the stage of hallucination. The model outputs become more erroneous, as they continued building on the errors of previous models.

5 Discussion

In this section, we compile our deductions and speculations based on all experiments conducted for sequential learning, model size, data complexity, overlap, and contradiction.

5.1 Does Model Size Matter?

How much can increasing the size of Transformer models enhance their performance across varied datasets? Our results show that larger Transformer models indeed have the potential to improve learning outcomes and generalize more effectively across diverse data [68]. But is there a ceiling to these improvements? Absolutely. As we scale up the model size, its ability to comprehend and learn from both songs improves, reaching enhanced performance levels up to a certain threshold[10]. This highlights the crucial role of model size in learning ability.

Yet, what happens as we continue to expand the model? Interestingly, we observe a plateau in performance gains. Why is this the case? It suggests that there are diminishing returns on making the models even larger[68]. Beyond a certain point, increasing the size no longer contributes to better performance, indicating that while model size is integral to handling complex learning scenarios, simply making the model bigger is not always better[68]. This plateau in performance stresses the need for careful consideration of how large to build our models to ensure optimal learning efficiency without unnecessary computational expenses.

5.2 The Impact of Data Complexity on Model Learning

How do Transformer models handle learning from simple and complex songs? We tried to answer this question by training models on pairs of songs (simple and complex songs). What becomes evident is that when these models attempt to learn information from two distinct songs at once, their ability to generalize from one to the other varies depending on complexity of the songs involved.

Does the simple and the complex nature between the two songs affect the model's learning efficiency? Absolutely. Our findings suggest that when the songs are simple, the models learn better. When the songs are complex, the models struggle more.

5.3 Overlap and Contradiction in Training Data

Another key finding of this study is the significant impact of song characteristics, such as overlap and contradiction, on the learning dynamics of the models. Our findings provide insights into how these elements influence the model's ability to perform well across different types of data.

5.3.1 Can Transformers Handle Non-Overlapping Data?

Our results show that models trained on completely different datasets, such as a song and its reversed version, struggle to apply what they have learned to new, unrelated data. Specifically, when a model was trained on just one song, it was almost unable to predict the sequences of the reversed song, showing almost zero accuracy. This suggests that Transformers have difficulty when there is no overlap in features between the training and testing datasets. However, mixing 50% of each song during training helped improve accuracy significantly, though it still did not reach the higher accuracy levels seen in a decision tree model.

In sequential learning; cases where one model passes on its learned songs to the next as training input, our deduction was further reinforced. The models were trained on both songs together, and those with no overlap or contradiction presented a greater challenge to the models. In addition to models achieving low initial accuracy, the sequential learning lead to more pronounced declines in prediction accuracy over time. They struggled to outperform model-chains trained on songs with moderate or higher levels of contradiction. This observation holds true in both when models sequentially learn from the complete song, and when models predict their own song from only a snippet of the song and pass on the information to the next model in the sequence.

5.3.2 What About Minor Contradictions?

When we introduced small differences in the training data, such as changing every tenth note, the models handled it much better. This setup, where the songs mostly overlapped but had slight differences, allowed the models to achieve better results. They could still use the common patterns to make accurate predictions. The best results arose from training the model equally with both the original and slightly changed songs, reaching accuracy levels comparable to the decision tree model. Similarly, models in sequential learning achieved the highest initial accuracy as well the slowest decline both on the complete song and a snippet of the song.

5.3.3 How Do Frequent Alterations Affect Learning?

A more challenging situation was when every fifth note in the song was changed. This led to a significant drop in accuracy, showing that more frequent changes make it harder for the model to learn and make correct predictions. However, training the model with equal parts of the original and heavily altered song did improve results considerably, achieving better accuracy than the decision tree model but not as high as when the changes were less frequent.

In contrast, sequential learning with the aforementioned song-pair rendered results which were not so different from the song-pair with minor contradictions. When learning

from the complete song, the decline in accuracy was noticeably faster than songs with minor contradictions, yet better than the song-pair with no overlaps/contradictions. When learning from a snippet of the song, the decline for the song-pair with high contradictions was closely similar in rate and speed to that with minor contradictions. The only noticeable difference was the initial model achieving a lower accuracy and the final model displaying a lower accuracy than models trained on the song-pair with minor contradictions.

5.4 Effectiveness of Training Paradigms

We explored two distinct training paradigms: learning from the complete song and building the song using only a snippet. In both scenarios, the dimension of "content pollution" was introduced differently. Learning from the complete song allowed models to iteratively refine their predictions by continuing the song note-by-note. While the initial model began training with the original song, the subsequent models trained on what each previous model "thought" the original song looked like. Thus, the original truth degraded gradually when a model predicted the next token of the sequence given the input of the previous model. Here, the model predicted the entire song at once, which requires it to capture longer-term dependencies and patterns across the entire dataset. This broader prediction scope may have increased the complexity of the task. It may have been the reason for the observed steeper decline in accuracy compared to predicting the song only from snippets at a time.

In contrast, building the song from snippets introduced a feedback loop wherein models generated sequences based on their own predictions. This, as we intuitively expect, should lead to a more pronounced decline in prediction accuracy. This is because the model constructs every subsequent prediction of the next "word" until it completes the song. It facilitates a larger deviation from the "original truth." However, its scope was limited in that it received one snippet, and had to predict only the next snippet at a time. This approach allows the model to focus on predicting shorter sequences. It may have potentially lead to more accurate predictions due to reduced complexity and shorter dependencies. We speculate that due to the aforementioned reasons, although we observed a decline in prediction accuracy in both predicting the entire next song and predicting the next snippet of the song, the former may have displayed a faster decline. The latter approach, however, invites a model to pollute its own content more frequently, while subsequent models do the same.

A model's ability to generalize well depends on how much the target data are represented in the source dataset [56]. As each sequential model produced its version of the source, the target data became progressively less represented. Therefore their ability to generalize declined. As more polluted content enter training datasets of LLMs, based on similar implications to our experiments, their ability to generalize well may decline.

5.4.1 Model Hallucination

Hallucination is a particularly intriguing phenomenon observed in our experiments. Models generated sequences that deviated significantly from the original data but maintained some degree of precision. As noted in "The Curse of Recursion: Training on Generated Data Makes Models Forget" [40], the more models iteratively generated sequences based on their own predictions, the more errors accumulated over time. This phenomenon parallels the hypothesis proposed by [80], who referred to it as the "hallucination snowballing" effect, where LLMs produce more hallucinations to remain consistent with earlier ones.

Our models did "forget" the original song, producing nonsensical patterns consistently. This aligns with the observations in "A Survey of Hallucination in Large Foundation Models" [64], which states that the patterns models produce appear plausible and coherent at first glance, yet deviate significantly from the base truth. This issue is also linked to the quality of training data [62], which, in our case, deteriorates as models introduce more errors through the cascade. In simple 'songs' like those in our dataset, this phenomenon might be relatively easy to identify. However, in large-scale models, hallucinatory behavior becomes more difficult to detect, especially if the output seems plausible enough for a human to mistakenly confirm it as accurate.

As more inaccuracies are introduced by models to their generated source for the next model, the more the quality of training data drops. While the initial model trained on the original uncontaminated truth, the models themselves corrupted their data later on. Consequently the model then over-fits those inaccuracies, leading to amplification of inaccuracies and hallucination [56].

5.5 Inference

On one hand, the results of our experiments show that transformers can adapt to and learn from data that has overlapping and contradicting elements. On the other hand, they also show that the extent of these contradictions and the presence of overlapping features greatly affect performance and the quality of model outputs. This applies to both standalone and sequential learning contexts.

Contradictions and overlaps are inherent qualities of a dataset, especially at the magnitude at which popular LLMs such as GPT-4 are trained [55]. Simply because books discussing capitalist ideologies and socialist ideologies contradict, it should not lead to the removal of one category or the other from the training dataset. Eliminating contradictions in such a manner might be counter-intuitive - it may lead to biased echo-chambers as opposed to the intended unbiased trained model.

Although we agree that careful dataset preparation can be an effective preemptive strategy for training future generative AI models, our study does not cover solutions for such dataset preparation. Rather, we use our experiments to highlight the consequences

of data overlap, contradictions, and models training on their own (erroneous) outputs.

5.6 Limitations of the study

Our study demonstrated that the size of the model, the complexity of the data, and the degree of overlap and contradiction within the training data significantly influence model performance and consistency. We acknowledge our findings within the bounds of the limitations mentioned below:

We used sequences of musical notes as the input for our Transformer models. Typically, Transformers are applied to textual data, often in the form of sentences. The Transformer model used in our experiments was a scaled-down version, referred to here as a 'toy model'. This smaller scale might not fully capture the capabilities and performance characteristics of larger, more complex Transformer models used in real-world applications.

Some of our experiments involved mixing fractions of two different songs to introduce data variability and complexity. This method, while useful for controlled experimentation, does not necessarily mimic the complex and dynamic data environments encountered in practical applications. Another set of our experiments examined two paradigms within which sequential model outputs degrade over time. While the implications are valid to LLMs on a conceptual level, extensive study on actual larger models may be needed to affirm our standpoint.

In our study, we focused on a specific type of overlap and contradiction using "musical" sequences. We looked at how identical input sequences from different songs produce the same or different output sequences. While this method is clear, easy to measure, and extend to further sequential data, its scope may be limited to less sophisticated contexts(textual contradiction). Overlap and contradiction can also be found in more complex areas such as political ideologies. Recall the example of liberalism and social democracy from section 1.4. They share overlapping ideas about democracy and individual rights, while capitalism and socialism hold contradicting views about property and the economy. These are broader and more nuanced contexts(contextual contradiction) which we did not explore in our study.

6 Conclusion

Through this study, we have explored the behavior of transformer models when faced with overlapping and contradictory data, as well as when they learn from other models. We demonstrated the significance of model size, data complexity, data overlap, and contradiction in model performance. We also identified a significant degradation of training data and model consistency when model-generated outputs entered their own or other models' training datasets. Our key observations are as follows:

6.1 Key Observations

1. **Impact of Sequential Learning:** We discovered that there are risks associated with using a model's own generated content for subsequent training. This practice can result in the propagation of repeated errors and may ultimately compromise the reliability of the information over time.

2. **Handling Contradictory and Overlapping Information:** Our results show that model size and data complexity plays a role in the learning ability of the model. Transformer models struggle with non-overlapping datasets but perform better with minor contradictions in the data. Training with a mix of songs improved accuracy. Sequential learning further highlighted these challenges, with non-overlapping data leading to significant declines in prediction accuracy over time. Conversely, models trained on data with minor contradictions maintained higher accuracy and exhibited slower declines. These findings underline the importance of data overlap and contradictions in learning capabilities of Transformer models.

In conclusion, recent advances in generative AI tools hold great potential to positively transform several industries. However, they also need careful management to prevent the spread of inaccurate information. It is understood that most Large Language Models are trained on publicly available internet data. Yet, there still remains a significant gap in our understanding of what happens when their own content or other AI-generated materials are added to training datasets. As the proportion of AI-generated content in these datasets grows, the distinction between human and AI contributions becomes increasingly blurred. This blend has the potential to erode the 'base truth,' compromising the foundational accuracy and quality of information. While phenomena like hallucination are already evident, the full extent of their impact on the model's learning outcomes and practical applications remains unclear.

By illustrating the emergence of these issues in a simpler version of the aforementioned models, we reveal certain challenges the larger, more complex models might face in the future.

6.2 Looking Ahead

Future research should prioritize enhancing the robustness of these models to the aforementioned challenges. While these AI models hold great potential, their effective utilization needs thoughtful management to contain the propagation of inaccurate information. This includes developing more refined methods for curating AI-generated content and enhancing the models' ability to navigate complex or conflicting data. Such endeavors are crucial to harnessing the full capabilities of Large Language Models while mitigating the associated risks. Our study contributes to starting a conversation about the future of AI in creating content, offering insights that will guide further research and policy decisions in this area.

References

[1] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. Chatterji, A. Chen, K. Creel, J. Q. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. Gillespie, K. Goel, N. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. Krass, R. Krishna, R. Kuditipudi, A. Kumar, F. Ladhak, M. Lee, T. Lee, J. Leskovec, I. Levent, X. L. Li, X. Li, T. Ma, A. Malik, C. D. Manning, S. Mirchandani, E. Mitchell, Z. Munyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J. C. Niebles, H. Nilforoshan, J. Nyarko, G. Ogut, L. Orr, I. Papadimitriou, J. S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. Roohani, C. Ruiz, J. Ryan, C. Ré, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. Srinivasan, A. Tamkin, R. Taori, A. W. Thomas, F. Tramèr, R. E. Wang, W. Wang, B. Wu, J. Wu, Y. Wu, S. M. Xie, M. Yasunaga, J. You, M. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, and P. Liang, "On the opportunities and risks of foundation models," 2022.

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.

[3] R. Dale, "Gpt-3: What's it good for?," *Natural Language Engineering*, vol. 27, pp. 113–118, 01 2021.

[4] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," 2023.

[5] Anthropic, "Meet claude-3: our latest clip-r model," 2023.

[6] T. Alqahtani, H. A. Badreldin, M. Alrashed, A. I. Alshaya, S. S. Alghamdi, K. bin Saleh, S. A. Alowais, O. A. Alshaya, I. Rahman, M. S. Al Yami, and A. M. Albekairy, "The emergent role of artificial intelligence, natural learning processing, and large language models in higher education and research," *Research in Social and Administrative Pharmacy*, vol. 19, no. 8, pp. 1236–1242, 2023.

[7] S. Pichai and D. Hassabis, "Our next-generation model: Gemini 1.5," 2024.

[8] J. A. Goldstein, G. Sastry, M. Musser, R. DiResta, M. Gentzel, and K. Sedova, "Generative language models and automated influence operations: Emerging threats and potential mitigations," 2023.

[9] M. Jakesch, J. T. Hancock, and M. Naaman, "Human heuristics for ai-generated language are flawed," *Proceedings of the National Academy of Sciences*, vol. 120, Mar. 2023.

[10] V. Ashish, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, p. I, 2017.

[11] M. A. K. Raiaan, M. S. H. Mukta, K. Fatema, N. M. Fahad, S. Sakib, M. M. J. Mim, J. Ahmad, M. E. Ali, and S. Azam, "A review on large language models: Architectures, applications, taxonomies, open issues and challenges," *IEEE Access*, vol. 12, pp. 26839–26874, 2024.

[12] S. Monteith, T. Glenn, J. R. Geddes, P. C. Whybrow, E. Achtyes, and M. Bauer, "Artificial intelligence and increasing misinformation," *The British Journal of Psychiatry*, vol. 224, no. 2, p. 33–35, 2024.

[13] L. Floridi and M. Chiriatti, "Gpt-3: Its nature, scope, limits, and consequences," *Minds and Machines*, vol. 30, no. 4, pp. 681–694, 2020.

[14] J. Gertner, "A.i.-generated content discovered on news sites, content farms and product reviews," *The New York Times*, July 2023. Retrieved 9/10/2023 from `https://www.nytimes.com/2023/07/18/magazine/wikipedia-ai-chatgpt.html`.

[15] Makyen, "Temporary policy: Generative ai (e.g., chatgpt) is banned." Forum post, Aug. 2023. Retrieved Sept. 12, 2023 from `https://meta.stackoverflow.com/q/421831/9737437`.

[16] A. Palmer, "People are using a.i. chatbots to write amazon reviews." CNBC, Apr. 2023. Retrieved Sept. 10, 2023 from `https://www.cnbc.com/2023/04/25/amazon-reviews-are-being-written-by-ai-chatbots.htm`.

[17] M. Sadeghi and L. Arvanitis, "Rise of the newsbots: Ai-generated news websites proliferating online." NewsguardTech, May 2023. Retrieved Sept. 10, 2023 from `https://www.newsguardtech.com/special-reports/newsbots-ai-generated-news-websites-proliferating/`.

[18] K.-C. Yang and F. Menczer, "Anatomy of an ai-powered malicious social botnet," 2023.

[19] J. Bright, F. E. Enock, S. Esnaashari, J. Francis, Y. Hashem, and D. Morgan, "Generative ai is already widespread in the public sector," 2024.

[20] P. Villalobos, J. Sevilla, L. Heim, T. Besiroglu, M. Hobbhahn, and A. Ho, "Will we run out of data? an analysis of the limits of scaling datasets in machine learning," 2022.

[21] G. Martínez, L. Watson, P. Reviriego, J. A. Hernández, M. Juarez, and R. Sarkar, "Towards understanding the interplay of generative artificial intelligence and the internet," 2023.

[22] G. Martínez, L. Watson, P. Reviriego, J. A. Hernández, M. Juarez, and R. Sarkar, "Combining generative artificial intelligence (ai) and the internet: Heading towards evolution or degradation?," 2023.

[23] J. Grimmelmann, *The virtues of moderation*, vol. 17. Yale JL & Tech., 2015.

[24] T. Lloyd, J. Reagle, and M. Naaman, ""there has to be a lot that we're missing": Moderating ai-generated content on reddit," 2024.

[25] C. Lampe and P. Resnick, "Slash(dot) and burn: distributed moderation in a large online conversation space," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*, (Vienna, Austria), pp. 543–550, Association for Computing Machinery, 2004.

[26] J. N. Matias, "The civic labor of volunteer moderators online," *Social Media + Society*, vol. 5, no. 2, p. 2056305119836778, 2019.

[27] B. Yu, J. Seering, K. Spiel, and L. Watts, ""taking care of a fruit tree": Nurturing as a layer of concern in online community moderation," in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA '20)*, (Honolulu, HI, USA), pp. 1–9, Association for Computing Machinery, 2020.

[28] K. Radivojevic, N. Clark, and P. Brenner, "Llms among us: Generative ai participating in digital discourse," 2024.

[29] S. A. Thompson and T. Hsu, "How easy is it to fool a.i.-detection tools?." The New York Times, June 2023. Retrieved Sept. 11, 2023 from `https://www.nytimes.com/interactive/2023/06/28/technology/ai-detection-midjourney-stable-diffusion-dalle.html`.

[30] E. Clark, T. August, S. Serrano, N. Haduong, S. Gururangan, and N. A. Smith, "All that's 'human' is not gold: Evaluating human evaluation of generated text," 2021.

[31] B. Dosono and B. Semaan, "Moderation practices as emotional labor in sustaining online communities: The case of aapi identity work on reddit," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*, (Glasgow, Scotland UK), pp. 1–13, Association for Computing Machinery, 2019.

[32] A. M. Schöpke-Gonzalez, S. Atreja, H. N. Shin, N. Ahmed, and L. Hemphill, "Why do volunteer content moderators quit? burnout, conflict, and harmful behaviors," *New Media & Society*, vol. 0, no. 0, 0.

[33] A. Loth, M. Kappes, and M.-O. Pahl, "Blessing or curse? a survey on the impact of generative ai on fake news," 2024.

[34] I. Solaiman, Z. Talat, W. Agnew, L. Ahmad, D. Baker, S. L. Blodgett, H. D. I. au2, J. Dodge, E. Evans, S. Hooker, Y. Jernite, A. S. Luccioni, A. Lusoli, M. Mitchell, J. Newman, M.-T. Png, A. Strait, and A. Vassilev, "Evaluating the social impact of generative ai systems in systems and society," 2023.

[35] F.-M. Vassel, E. Shieh, C. R. Sugimoto, and T. Monroe-White, "The psychosocial impacts of generative ai harms," 2024.

[36] V. Capraro, A. Lentsch, D. Acemoglu, S. Akgun, A. Akhmedova, E. Bilancini, J.-F. Bonnefon, P. Brañas-Garza, L. Butera, K. M. Douglas, J. A. C. Everett, G. Gigerenzer, C. Greenhow, D. A. Hashimoto, J. Holt-Lunstad, J. Jetten, S. Johnson, C. Longoni, P. Lunn, S. Natale, I. Rahwan, N. Selwyn, V. Singh, S. Suri, J. Sutcliffe, J. Tomlinson, S. van der Linden, P. A. M. V. Lange, F. Wall, J. J. V. Bavel, and R. Viale, "The impact of generative artificial intelligence on socioeconomic inequalities and policy making," 2024.

[37] J. Bullock and M. Luengo-Oroz, "Automated speech generation from un general assembly statements: Mapping risks in ai generated texts," 2019.

[38] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen, "A survey of large language models," 2023.

[39] S. Alemohammad, J. Casco-Rodriguez, L. Luzi, A. I. Humayun, H. Babaei, D. LeJeune, A. Siahkoohi, and R. G. Baraniuk, "Self-consuming generative models go mad," 2023.

[40] I. Shumailov, Z. Shumaylov, Y. Zhao, Y. Gal, N. Papernot, and R. Anderson, "The curse of recursion: Training on generated data makes models forget," 2024.

[41] J. Hartmann, J. Schwenzow, and M. Witte, "The political ideology of conversational ai: Converging evidence on chatgpt's pro-environmental, left-libertarian orientation," 2023.

[42] J. Joiner, M. Piva, C. Turrin, and S. W. C. Chang, "Social learning through prediction error in the brain," *npj Science of Learning*, vol. 2, no. 1, p. 8, 2017.

[43] J. S. Bruner, "The act of discovery," *Harvard Educational Review*, vol. 31, no. 1, pp. 21–32, 1961.

[44] M. Csikszentmihalyi, *Creativity: Flow and the psychology of discovery and invention*. Harper Perennial, 1996.

[45] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *Behavioral and Brain Sciences*, vol. 40, 2017.

[46] A. Jordanous, "A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative," *Cognitive Computation*, vol. 8, no. 3, pp. 319–332, 2016.

[47] openAI, "Customizing gpt-3 for your application," 2021. 2024.

[48] N. Jain, K. Saifullah, Y. Wen, J. Kirchenbauer, M. Shu, A. Saha, M. Goldblum, J. Geiping, and T. Goldstein, "Bring your own data! self-supervised evaluation for large language models," 2023.

[49] J. Devlin and et al., "Bert: Bidirectional encoder representations from transformers," *arXiv preprint arXiv:1810.04805*, 2018.

[50] A. Radford and et al., "Improving language understanding by generative pre-training," *URL https://openai. com/research/*, 2018.

[51] G. Sachs, "Generative ai could raise global gdp by 7%," 2023.

[52] F. Alahdab, "Potential impact of large language models on academic writing," *BMJ Evidence-Based Medicine*, 2023.

[53] S. Feuerriegel, J. Hartmann, C. Janiesch, and P. Zschech, "Generative ai," *Business & Information Systems Engineering*, vol. 66, no. 1, pp. 111–126, 2024.

[54] C. Liu, A. Shah, W. Bai, and R. Arcucci, "Utilizing synthetic data for medical vision-language pre-training: Bypassing the need for real images," 2024.

[55] P. Tiwald, A. Ebert, and D. T. Soukup, "Representative and fair synthetic data," 2021.

[56] T. Alkhalifah, H. Wang, and O. Ovcharenko, "Mlreal: Bridging the gap between training on synthetic data and real data applications in machine learning," *Artificial Intelligence in Geosciences*, vol. 3, pp. 101–114, 2022.

[57] S. Overflow, "Temporary policy: Chatgpt is banned," 2022.

[58] G. Spitale, N. Biller-Andorno, and F. Germani, "Ai model gpt-3 (dis)informs us better than humans," *Science Advances*, vol. 9, June 2023.

[59] N. Köbis and L. D. Mossink, "Artificial intelligence versus maya angelou: Experimental evidence that people cannot differentiate ai-generated from human-written poetry," *Computers in Human Behavior*, vol. 114, p. 106553, 2021.

[60] S. Kreps, R. M. McCain, and M. Brundage, "All the news that's fit to fabricate: Ai-generated text as a tool of media misinformation," *Journal of Experimental Political Science*, vol. 9, no. 1, p. 104–117, 2022.

[61] C. Metz, C. Kang, S. Frenkel, S. A. Thompson, and N. Grant, "How tech giants cut corners to harvest data for a.i.," April 2024. Accessed: 2024-05-02.

[62] X. Guo and Y. Chen, "Generative ai for synthetic data generation: Methods, challenges and the future," 2024.

[63] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," 2023.

[64] V. Rawte, A. Sheth, and A. Das, "A survey of hallucination in large foundation models," 2023.

[65] Z. Xu, S. Jain, and M. Kankanhalli, "Hallucination is inevitable: An innate limitation of large language models," 2024.

[66] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," 2023.

[67] B. Buchanan, A. Lohn, M. Musser, and K. Sedova, "Truth, lies, and automation," *Center for Security and Emerging technology*, vol. 1, no. 1, p. 2, 2021.

[68] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, 2020.

[69] C. R. Harris, K. J. Millman, S. J. Van Der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, *et al.*, "Array programming with numpy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.

[70] R. Length, "War and peace by leo tolstoy," 2023.

[71] A. Semenov, V. Boginski, and E. L. Pasiliao, "Neural networks with multidimensional cross-entropy loss functions," in *Computational Data and Social Networks: 8th International Conference, CSoNet 2019, Ho Chi Minh City, Vietnam, November 18–20, 2019, Proceedings 8*, pp. 57–62, Springer, 2019.

[72] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[73] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception*, pp. 65–93, Elsevier, 1992.

[74] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade: Second Edition*, pp. 421–436, Springer, 2012.

[75] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[76] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information processing & management*, vol. 45, no. 4, pp. 427–437, 2009.

[77] D. J. Sheskin, *Handbook of parametric and nonparametric statistical procedures*. Chapman and hall/CRC, 2003.

[78] B. Charbuty and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20–28, 2021.

[79] R. Geirhos, C. R. Temme, J. Rauber, H. H. Schütt, M. Bethge, and F. A. Wichmann, "Generalisation in humans and deep neural networks," *Advances in neural information processing systems*, vol. 31, 2018.

[80] M. Zhang, O. Press, W. Merrill, A. Liu, and N. A. Smith, "How language model hallucinations can snowball," 2023.