WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

EQUIS ACCREDITED AACSB ACCREDITED ASSOCIATION AMBA ACCREDITED

# Deep Generative Models for Synthetic Data

Eigenschink, Peter; Vamosi, Stefan; Vamosi, Ralf; Sun, Chang; Reutterer, Thomas; Kalcher, Klaudius

*DOI:*
10.1109/ACCESS.2023.3275134

Published: 01/01/2021

*Document Version*
Early version, also known as preprint

Link to publication

*Citation for published version (APA):*
Eigenschink, P., Vamosi, S., Vamosi, R., Sun, C., Reutterer, T., & Kalcher, K. (2021). *Deep Generative Models for Synthetic Data*. https://doi.org/10.1109/ACCESS.2023.3275134

Download date: 12. Oct 2024

# DEEP GENERATIVE MODELS FOR SYNTHETIC DATA

**Peter Eigenschink**
Department of Marketing
Vienna University of Economics and Business
Vienna, Austria
peter.eigenschink@wu.ac.at

**Stefan Vamosi**
Department of Marketing
Vienna University of Economics and Business
Vienna, Austria
stefan.vamosi@wu.ac.at

**Ralf Vamosi**
Department of Marketing
Vienna University of Economics and Business
Vienna, Austria
ralf.vamosi@wu.ac.at

**Chang Sun**
Institute of Data Science
Maastricht University
Maastricht, The Netherlands
chang.sun@maastrichtuniversity.nl

**Thomas Reutterer**
Department of Marketing
Vienna University of Economics and Business
Vienna, Austria
thomas.reutterer@wu.ac.at

**Klaudius Kalcher**
Mostly AI GmbH
Vienna, Austria
klaudius.kalcher@mostly.ai

November 23, 2021

**ABSTRACT**

Growing interest in synthetic data has stimulated development and advancement of a large variety of deep generative models for a wide range of applications. However, as this research has progressed, its streams have become more specialized and disconnected from each other. For example, models for synthesizing text data for natural language processing cannot readily be compared to models for synthesizing health records. To mitigate this isolation, we propose a data-driven evaluation framework for generative models for synthetic data based on five high-level criteria: *representativeness*, *novelty*, *realism*, *diversity* and *coherence* of a synthetic data sample relative to the original data-set regardless of the models' internal structures. The criteria reflect requirements different domains impose on synthetic data and allow model users to assess the quality of synthetic data across models. In a critical review of generative models for sequential data, we examine and compare the importance of each performance criterion in numerous domains. For example, we find that realism and coherence are more important for synthetic data for natural language, speech and audio processing, while novelty and representativeness are more important for healthcare and mobility data. We also find that measurement of representativeness is often accomplished using statistical metrics, realism by using human judgement, and novelty using privacy tests.

***K*eywords** Artificial intelligence, neural networks, deep learning, generative models, synthetic data, sequential data, big data, privacy

## 1 Introduction

In recent years, adoption of deep generative models for synthetic data have spread to a variety of domains. Such models can generate impressive synthetic images [1], text [2], and music [3] as well as sensory data [4], electronic health records [5], mobility trajectories [6], and financial time-series [7]. This significant progress was made possible by accessibility of vast amounts of data and computing technologies capable of handling the data, both connected to the

rise of "big data" and advances in deep learning. Models based on deep learning can handle large amounts of complex, highly correlated, high-dimensional data and generate synthetic data for many use-cases. With the capacity have come challenges that have also led to increase in methodological advances in fields in which synthetic data have been applied. In particular, models for generating synthetic data boosted progress in data augmentation [8], data imputation [9], fairness in biased data-sets [10], and sharing of privacy-sensitive data-sets [11]. Today, deep generative data synthesis is a large and mature field that involves many streams of research across a wide range of domains.

Overall, the field has advanced in big leaps, but research in various (sub-) domains has tended to drift apart. Indeed, it is difficult to compare models applied to problems in natural language processing (NLP) with models for the generation of synthetic health records. Still, some domains share common characteristics, and models applied in one field can be applied in others. Consider, for example, the recent success of so-called transformer models introduced in natural language generation (NLG) [12, 13] and now being applied in other domains to generate synthetic time-series data [14]. Because it is not always possible to transfer models to other fields, new insights can remain isolated to specific domains and fail to disseminate. The two most common barriers are (i) heterogeneity of the data and (ii) conflicting requirements for synthetic data in different use-cases.

Because research in one domain can benefit from insights from other domains, we need a common basis for discussing generative models and guiding research, especially in domains in which research to date is sparse.

To facilitate this discussion, we propose a framework for deep generative models designed to generate synthetic data based on high-level evaluation criteria. This framework addresses the barriers of heterogeneity in the data and the data requirements via abstraction and allows researchers to put generative models into broader contexts. We present a critical review of publications on deep generative models in the context of synthetic sequential data and apply the proposed framework to those models.

We chose to focus on sequential data for two specific reasons. First, dynamic phenomena are particularly relevant in many fields and pose significant challenges for modelers and analysts. Examples of uses of sequential data are geo-locations [15]; shopping paths [16]; text [2]; videos [17]; music [3]; behavior in digital environments such as, music [18] and video streaming [19], clickstreams [20], and internet browsing [21]; financial transactions [22]; and electronic health records [5]. All of these types of data share some underlying correlational structures within sequences but also are heterogeneous in terms of the dimensionality and cardinality of steps in a sequence. Our second reason for focusing on the evaluation of synthetic sequential data generated by deep learning models is that it nicely complements previous reviews in related fields, such as reviews of deep learning [23] and architectures of deep generative models [24, 25, 26]. Furthermore, a number of review articles have focused on specific model architectures, such as generative adversarial networks [27], normalizing flows [28], and synthetic data in specific domains such as, molecular science [29] and graph data [30]. The scope of those articles is narrow; they address specific domains and the disregard literature on other data (see, for example, [31] on music and [32] on text). Thus, we aim to fill the gap between general and specific reviews of deep generative models for sequential data.

The next section introduces our high-level evaluation framework for generative models. Then, in section 3, we give a brief overview of popular deep learning architectures used to generate synthetic sequential data. In section, 4 we review applications of synthetic data in different domains, compare strengths and weaknesses of the used models and their architectures and critically reflect them according to the framework we propose in section 2.

## 2 Evaluation of generative models for synthetic data

Metrics to evaluate the performance of deep generative models are as diverse as the models' objectives and specific data structures involved. General-purpose metrics, such as the commonly used negative log-likelihood (NLL), average log-likelihood (ALL) and maximum mean discrepancy (MMD) are rare and have limitations of their own [33]. Other metrics are specific particular model architectures. [34, 35], for example, gives a thorough overview of metrics commonly used to evaluate generative adversarial networks (GANs). Some metrics are domain-specific, such as the classifier-based inception score (IS) for synthetic images proposed by [36]. [33] reviews metrics used to evaluate generative models in the visual domain, and [30] for graph data. These metrics are effective for measuring progress in specific domains and to compare models of a specific type, such as GANs; using them to compare different models and domains can be challenging. Even when considering only sequential data, heterogeneity is quickly apparent. What such heterogeneous data have in common are patterns of serial correlations within sequences. The cardinality and dimensionality of the data illustrate its heterogeneity, being augmented only further by the lengths of sequences. For example, text is one-dimensional and discrete since it is made up of single words in a discrete vocabulary. Video data, on the other hand, is continuous and high-dimensional. At each step, there is a whole image that consists of many pixels, each is described by real numbers between $0$ and $255$. Figure 1 illustrates the heterogeneity in the landscape of

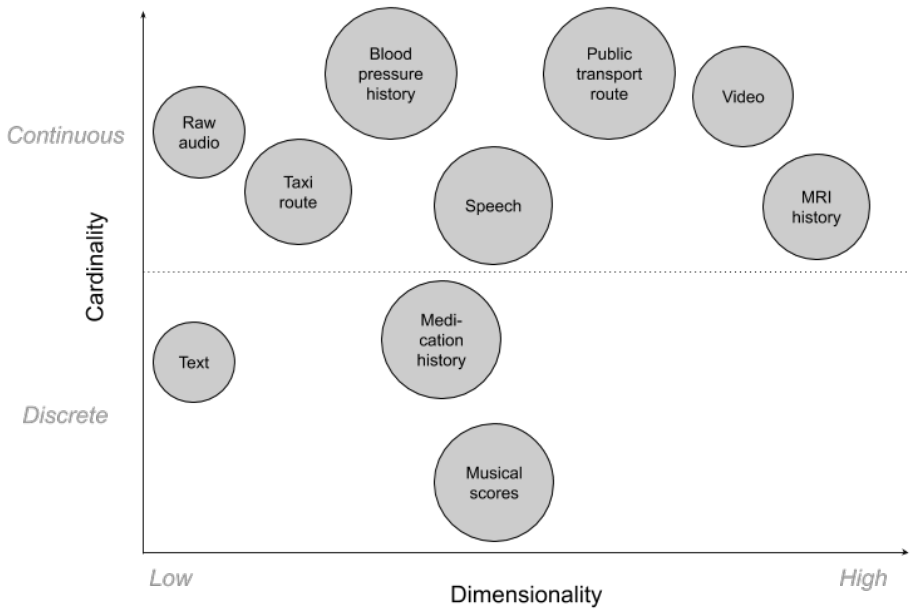


Figure 1: An illustration of the heterogeneity of sequential data based on cardinality and dimensionality.

sequential data by plotting the cardinality and dimensionality the data for several examples of sequential data relative to each other.

To tackle the numerous challenges associated with heterogeneous data and applications, we propose five high-level abstract criteria for evaluation of generative models: *representativeness*, *novelty*, *realism*, *diversity*, and *coherence*. The criteria are designed to compare the original data to the synthetically generated sample and can be applied to any generative model for synthetic data (see [37] for a recent example of a holdout-based framework for empirical assessment). They reflect requirements that are imposed on synthetic data in specific use-cases.

Because the criteria can be imposed on numerous types of sequential data, obtaining high scores on all five will rarely be the goal. Borji reviews qualitative and quantitative metrics for generative models in [34, 35], but there is no one-to-one mapping between those criteria and ours. The two approaches share some aspects, reflected in what [34] defined as the desiderata of evaluation measures.

Our proposed criteria are abstract in nature but capture different concrete metrics depending on the use-case. Furthermore, some of our criteria conflict with each other. For example, we expect to see trade-offs between high representativeness of the synthetic data-set and novelty. Figure 2 illustrates synthetic data that have high and low scores on each criterion relative to a given data-set.

### 2.1 Evaluation Criteria

#### 2.1.1 Representativeness

The representativeness of a generative model for synthetic data describes its ability to capture population-level properties of the original data. Ideally, generative models distill abstract structures from a set of training data. Consequently, the population-level properties of the synthetic and original data should be the same. For example, a data-set of face images is likely to have a certain distribution of hair colors and eye distances, and those distributions and the dependencies between the distributions (e.g., gray hair and the amount of wrinkles on a face) in the original and synthetic data should match. Depending on the type of data, there can be a multitude of ways to measure and quantify the similarity of the distributions.

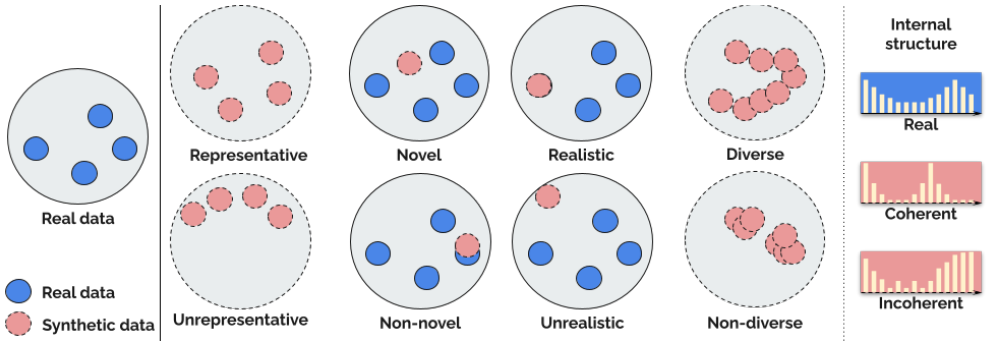


Figure 2: Illustration of synthetic data-sets that score high (top) and low (bottom) on the proposed criteria when compared to the original data-set on the left. *Coherence* only captures the internal structure of the data and is illustrated on the right.

Representativeness of synthetic data matters because statistical analyses and machine learning methods performed on synthetic data should result in the same statistical findings as analysis of the original data. A lack of representativeness despite all other criteria being fulfilled, indicates that the synthetic data provide a good representation only of a biased subspace of the actual data distribution and miss potentially critical information.

In many cases, representativeness is evaluated by statistical measures. Common methods are ALL, MMD and Kullback-Leibler divergence (KLD), which compare the probability distribution of the original data to the approximation of the distribution by the generative model. Recently, representativeness has also been evaluated by comparing the performance of classification models applied to the original and the synthetic data (see [38] for an example in healthcare).

### 2.1.2 Novelty

Evaluating the novelty of data from a generative model compares the original and synthetic data at an individual level. Novelty is sometimes overlooked in explicit quality evaluations, but the value of synthetic data without novelty is typically quite limited. The goal of using deep generative models usually is creation of entirely novel data-points. Novelty means that the synthetic data-points are entirely new observations of the latent distribution of the original data and should not closely resemble any original data-points.

Models that generate only novel data-points do not allow any individual-level information from the training data to leak into the synthetic data. Thus, novelty is tightly linked to privacy, and a high novelty score indicates that the "inspiration" behind the synthetic data-points is not identifiable at the individual level. The synthetic data records could just as well have been a holdout subset of the original data. The opposite of high novelty is a model that memorizes and exactly recreates the training data. Such synthetic data would fulfill the other four criteria (since a copy of the original training data is obviously indistinguishable in many respects from that data).

In some cases, such as in NLP, novelty of the synthetic data is irrelevant. In other cases, however, such as creative domains (e.g., music composition), the goal is to generate new creative content. For example, [39] used the average Euclidean distance of a synthetic data-point from its nearest neighbor in the original data-set to measure the novelty of synthetic music (see Section 4.2 for details). In other cases, such as healthcare, privacy is more important than novelty. The generative models used to produce private synthetic data must not leak any sensitive information (see Section 4.4 for more details).

### 2.1.3 Realism

When considering an individual synthetic data-point generated by a highly realistic model on its own, it is difficult to know whether it is synthetic or original. Realism is similar to representativeness of the data, but at the individual subject level. A synthetic data-set can match all the statistics of the original data and still be unrealistic when individual data-points share characteristics that make them easily identifiable as synthetic. Consider, for example, a representative but unrealistic example obtained using a GAN trained on random cat images from the internet. Synthetic cat images can contain captions reminiscent of online memes that look plausible from a distance but actually consist solely of abstract symbols having shapes similar to letters.

Realism has been addressed in many publications in a variety of ways. The most common method is judgement of realism of the synthetic data by humans, either qualitatively (e.g. [40, 41]) or using empirical evaluations (e.g.

[42, 43, 44]). Evaluation studies present individuals with the original data-point and the synthetic data-point and ask them to choose which is the most. In some publications, participants in the evaluation studies were restricted to experts (e.g., medical experts in [45, 43] and music experts [46, 44]). In some cases, realism is quantitatively evaluated using objective measures. These evaluations are usually domain-specific and use metrics such as IS [47, 48] and the evaluation of synthetic music against theoretical music rules [49].

#### 2.1.4 Diversity

While representativeness, novelty and realism capture similarities between the original and the synthetic data, diversity measures similarities between each synthetic data-point and the whole synthetic data-set at the individual level. Therefore, models that score well on diversity generate unique data-points even when data-sets are large. Models that generate the same individual points over and over, such as some early versions of GANs, obviously lack diversity. For instance, generators sometimes create a single image that the discriminator cannot distinguish from an original image. Generating only that image is a local optimum, and the resulting effect is called mode collapse (see Section 3.6).

Many publications have not addressed the diversity of the generative models' synthetic data. In most cases, it is important that models achieve at least some diversity, and some models can generate only a small number of different samples (e.g., the aforementioned GAN with mode collapse).

There are several ways to measure the diversity of a model. Donahue et al. [39] used the average Euclidean distance of synthetic data-points to their respective nearest neighbors to evaluate the diversity of their model (see Section 4.2). Others have used metrics based on classifiers. For example, to measure the diversity of their video-generation models, the authors of [47] and [48] used the IS [36] (see Section 4.3). In other cases diversity has been captured only by subjective qualitative evaluations by humans.

#### 2.1.5 Coherence

Unlike the first for criteria, which are based on the structure of the synthetic data at an individual or population level, coherence captures the internal structure of single synthetic data-points, specifically their consistency. Coherence is particularly relevant for sequential data, that reflect sequential orders of events and for data such as images. Coherence requirements depend on the use-case and original data and can differ in terms of coarseness. For example, music should sound smooth and natural note-by-note and measure-by-measure, but also should stay within a certain genre overall. In images when multiple objects cast shadows from a single light source, the shadows must be coherent in terms of the direction in which they point and their length. While some incoherence in the data can lead to greater novelty or diversity, too much results in unrealistic data. For example, a music sample that frequently changes its genres would certainly sound creative but also would sound unrealistic.

Some studies have measured the coherence of synthetic data implicitly when evaluating its realism. In [46], for example, experts evaluated the naturalness of transitions in synthetic music. In many cases, however, domain-specific objective metrics have been used to judge coherence. In [48] coherence was computed using the average content distance between frames in synthetic videos (see Section 4.3).

## 3 Generative architectures for sequential data

Before we discuss the relevance of our proposed criteria for assessing the quality of synthetic data in various domains, we briefly introduce the architectural elements most frequently used in deep learning models for generating synthetic sequential data. Figure 3 provides an overview of the popularity of these architectural elements based on our review of articles reviewed for the current study.

At its core, synthetic data generation amounts to sampling from a distribution over a complex, high-dimensional data space. In addition to complexity arising from high dimensionality, the distribution is observed only indirectly via a set of training examples that only sparsely cover the space. Nonetheless, early work on generative deep learning models has already successfully dealt with both of these issues, thanks in part to transfers of proven architectural elements from other tasks applied to similar types of data. For instance, one of the earliest notable successes of deep-learning-based synthetic data generation was the creation of images using GANs. The adversarial networks were based on convolutional neural networks (CNNs), which were already well established for deep learning on images at this point.

Along similar lines, generative networks for sequential synthetic data were based on elements already successfully used for other deep learning tasks and sequential data of a similar type. Besides CNNs, these include recurrent neural networks (RNNs) and attention mechanisms, which are covered in the first half of this section. In addition to challenges associated with structuring of the network layers when synthesizing synthetic data is how to actually perform sampling

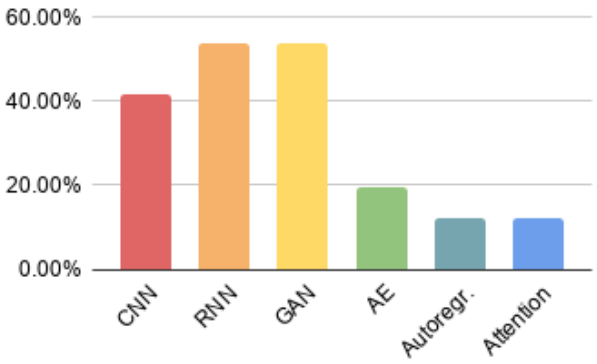


Figure 3: Popularity of various architectural elements in deep learning models used to generate synthetic sequential data. The graph shows the percentage of each architecture found to be the basis for models used in the reviewed studies. Note that a single model can be based on multiple architectures.

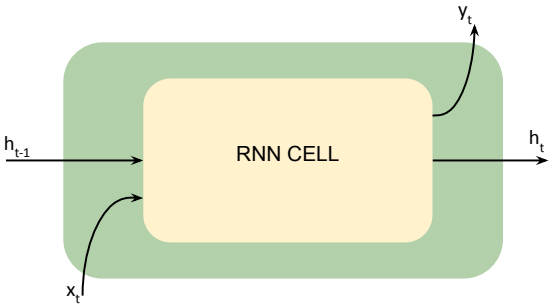


Figure 4: General principle of an RNN. In each time step, one RNN cell uses the input $x_t$ at that time step and the previous time step's hidden state $h_{t-1}$ to compute the new hidden state $h_t$ and output $y_t$. It is possible to stack multiple RNN layers by using the sequence of outputs of one layer as the input sequence for the next one.

in the data space. At a high level, two main approaches are used: (i) estimating the distribution explicitly and sampling from it (usually iteratively, as in autoregressive networks) and (ii) sampling from more straightforward distributions and mapping from that space to the more complex data space (autoencoders (AEs), GANs). These approaches are covered in detail in the second half of this section.

### 3.1 Recurrent Neural Networks

The network architecture most strongly associated with sequential data is RNNs, which are defined by sequential connections of cells that each take, as inputs, one element of the sequence and some output of the cell that processed the previous element in the sequence [50]. In this way, a chain of RNN cells can process sequences of arbitrary lengths (see Figure 4 for a sample schematic). The information passed from one cell to the next is called the hidden state of the network, and various types of RNNs can be defined by how this state is computed.

One of the most widespread types of RNNs is the long short-term memory (LSTM) network invented by Hochreiter and Schmidhuber [51] and improved by introduction of a "forget gate" by [52] to better handle long-term dependencies. See Figure 5 for a full schematic of this most canonical definition of the LSTM. Though extensions and variations have been introduced, this plain LSTM is the cornerstone for processing sequential data. It has been used with speech recognition [53], language translation [54], and natural language modeling [55].

Gated recurrent units (GRU) represent a relatively recent development in the RNN class. They were introduced in 2014 by Kyunghyun Cho et al. [56] and were inspired by LSTMs, with which they have much in common. They are characterized by requiring a smaller number of trainable parameters, because they have only three trainable layers per cell. Though GRUs perform similarly or slightly better than LSTMs in many applications involving relatively small data-sets, GRUs cannot perform unbounded counting. Thus, LSTM networks outperform GRU cells complex tasks including neural machine translation [57].

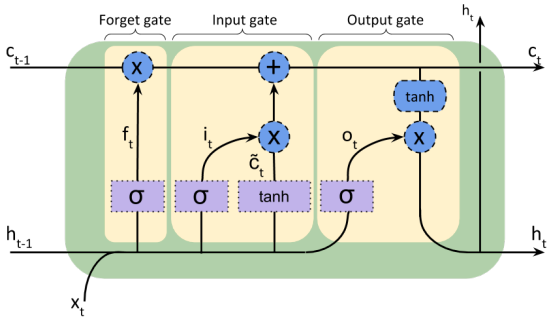


Figure 5: Diagram of an LSTM cell. Information passed from one time step to the next consists of a hidden state, $h$, that is equivalent to that cell's output and the cell state $c$ responsible for longer-term information storage. There are four trainable layers in each cell. The first is found in the forget gate and is used to determine any information in the cell state to be "forgotten" by multiplying it with the output of a sigmoid activation function (thus, between 0 and 1). The input gate is composed of two trainable layers and defines the new information to be written in the cell state. The final layer is in the output gate, which, together with the cell state, is used to compute the output of the cell.

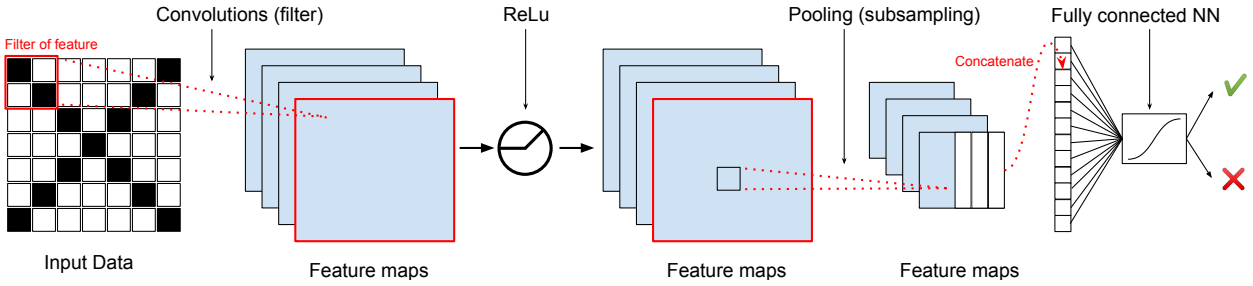


Figure 6: Structure of a CNN deciding whether an image belongs to a certain category. High-level features of the image are extracted via feature maps and are then pooled together. Finally, a fully connected layer decides whether the image belongs to the category based on the high-level features.

In the context of sequence generation, RNNs are most often employed in autoregressive architectures (see 3.4) and are trained by having them predict the next element of the input. Then, the trained layers are used to generate new sequences one step at a time. It is also common to use RNNs as elements of autoencoders (see 3.5). The hidden state is used as the bottleneck such that new sequences can be sampled by initializing predictions of a sequence with a seed in the latent space. Some GAN architectures have used RNNs, but it is not a common combination.

While RNNs can be powerful tools in sequence generation, they have limitations. Though the span of long-range dependencies they can capture is theoretically unlimited, RNNs can suffer from vanishing gradients and practical limitations regarding extremely long-range dependencies [58]. Furthermore, the computational effort required by RNNs can be intense, and RNNs do not scale well as sequence length increases.

### 3.2 Convolutional Neural Networks

Inspired by functioning of the visual cortex in mammals' brains, CNNs are one of the earliest modern deep learning patterns to be formalized [59]. They were instrumental to breakthroughs in deep learning for computer vision [60], used to solve tasks such as recognition of handwriting [61], faces [62] and documents [63]. Despite their origin, CNNs are not limited to images. They have been successfully applied to sequences and generative modeling.

The basic building block of a CNN are convolutional layers that consist of filters of equal weights being applied repeatedly to different segments of the data. The weights in the filters converge so each filter learns to compute particular features of the data without requiring explicit feature engineering beforehand. When one combines multiple layers together, the layers compute abstract features from low-level features like edges to high-level features such as facial patterns. Convolutional layers usually are connected using pooling layers that summarize and reduce the dimensionality of the previous layer by, for example, outputting the maximum of multiple input neurons. The last convolutional layer is then typically connected to a fully connected layer that takes the higher-level features as input to compute the desired information, such as classifying text. See Figure 6 for an illustration of the structure of a CNN

used with images. The result of this process is a neural network that is more resistant to noise than other architectures and invariant under spatial translation. The CNN uses a smaller number of trainable parameters and thus requires less memory and computational power, making it typically faster to train than RNNs.

These attributes make CNNs attractive for other tasks, including generative modeling of sequential data using convolutions on one dimension instead of two. However, causality constraints on many tasks in this area can require setting up the model so that each step depends on only past steps in the sequence in the previous layer, a pattern known as a temporal convolutional network (TCN) [64]. TCNs coupled with dilated convolutions [65, 40] have successfully captured long-term dependencies and proven successful in tasks as varied as weather prediction [66], traffic prediction [67], and in natural language processing [68, 69, 70].

When using CNNs to generate synthetic sequences, two approaches can be taken. The first is to predict the time steps one-by-one autoregressively (see Section 3.4), as demonstrated most prominently by WaveNet's text-to-speech approach [40]. Alternatively, deconvolutional neural networks [71] can be used to generate whole sequences at once based on a seed. Deconvolutional networks can be applied when using CNNs as part of an autoencoder or GAN (see Sections 3.5 and 3.6). For a recent survey of CNNs see [72].

### 3.3 Attention

The attention mechanism is a technique developed to improve sequence-to-sequence models by changing how information is transferred from the input sequence to the output sequence [73]. Instead of a strict encoder-decoder model in which an input sequence $(x_1, ..., x_n)$ is staticly mapped to a latent representation $z$, before $z$ is decoded to generate the output sequence $(y_1, ..., y_m)$, the attention mechanism allows the network to learn which parts of the input are relevant for specific parts of the output and to use information more selectively.

Attention layers were initially developed to complement RNNs and address one of their core limitations. In RNN networks, information that must be memorized over long stretches of a sequence being processed must be stored (and left untouched) in the cell state over numerous time steps, leading to information potentially being lost when updates delete or overwrite the information. The capacity of the cell state is limited and the model must decide which information to store and which information to replace without knowing which information will be needed later. The attention mechanism instead retrieves information from the past based on relevance, thus keeping the depth of back-propagation low compared to RNNs with no attention mechanism.

Today, attention layers are also used independently in so-called transformer networks [74] that consist of of layers of self-attention (i.e., inputs are copies of a single sequence instead of multiple sequences). The attention mechanism has also been successfully integrated in different types of networks. For example, Zhang et al. added an attention mechanism to their self-attention GAN [75] to improve the quality of generated images. The attention mechanism allows both the generator and the discriminator to connect image regions that are far apart, something CNNs are usually not capable of doing because of their focus on local properties (see Section 3.2). For a thorough introduction to attention mechanisms and review of their applications see [76].

### 3.4 Autoregressive Neural Networks

Autoregressive neural networks (AR-NNs) were the earliest network architecture used for sequence generation. These networks use the same data as input and output and thus predict the future of a sequence based on its past. Generally speaking, any network that models conditional probabilities ($p(x_i|x_1, x_2, ..., x_{i-1})$) can be called autoregressive, but the term is sometimes used specifically for networks that do not include any of the types of layer previously mentioned (RNNs, TCNs, transformers). Such networks sometimes only use dense layers to connect various time steps. For our purposes, we use the word in its wider sense as a generative strategy that is distinct from autoencoders and GANs. A distinguishing feature of the AR-NN strategy is that randomness in the input is not required to generate sequences. Rather, because the model predicts conditional probabilities, randomness is introduced immediately before the output layer by drawing from the probabilities.

The inner structure of AR-NNs as proposed by Bengio and Bengio in [77] is illustrated in Figure 7. Prediction for the $i$-th variable $x_i$ depends on the previously seen $i - 1$ inputs. Thus, the functions of the hidden layer are trained step-by-step, but are reused for subsequent variables $x_{i+1}, x_{i+2}, ....$

One popular autoregressive approach is the neural autoregressive density estimator (NADE) [78]. It uses the same principle but incorporates a weight sharing scheme (inspired by restricted Boltzmann machines) to improve generalization performance. Autoregressive networks can incorporate all of the structural building blocks previously described (RNNs, TCNs, transformers) as illustrated by autoregressive CNNs applied to pixel prediction [79] and sound generation [40].

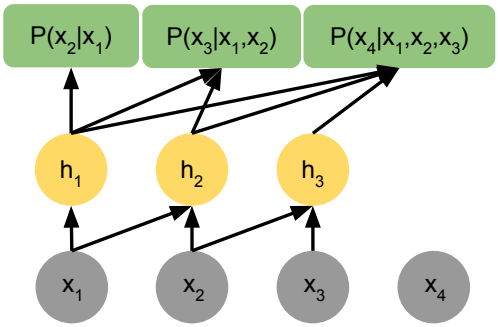


Figure 7: Simple example of an AR-NN based on [77]. This network uses dense layers to predict conditional distributions of later time steps based on previous ones.

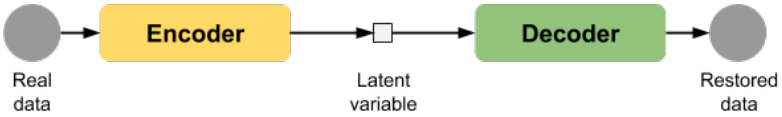


Figure 8: Architecture of an autoencoder. The encoder maps the input to a lower-dimensional representation. The decoder tries to optimally restore the input from that latent representation.

### 3.5 Autoencoders

Like autoregressive networks, autoencoders use the same data as input and output, but use a different strategy to prohibit the network from learning the trivial solution of simply passing through the full input. While autoregressive networks use a time lag to hide the critical bit of information (the last time step whose probability distribution is predicted), AEs feed the full input sequence into the model and use a bottleneck to make it impossible to pass through the full information from the input. Structurally, AEs can be split into encoders and decoders that designate the steps before and after the bottleneck, respectively. The encoder maps the input to an internal, lower-dimensional representation in a latent space in the bottleneck layer, and the decoder tries to restore the original data from that lower-dimensional representation. The loss function used to train the network is the dissimilarity between the input and output. See Figure 8 for an illustration of the architecture.

To create new data, the trained decoder is used in isolation with random values in the latent space as input. Crucially, the quality of the results depends on the structure and regularity of the latent space, and irregularities result in two common problems. First, points sampled from the latent space can lead to meaningless data once decoded, especially if the encoder leaves gaps in the latent space that are never covered by the original input data, forcing the decoder to extrapolate. Second, two points that are close together in the latent space can result in significantly dissimilar outputs in the original data space if the latent space is not smooth enough.

Variations of the AE have been proposed to overcome these problems that construct a latent space that has an adequate degree of regularity and typically forces the latent distribution to match a certain prior distribution [80, 81, 82, 83]. The

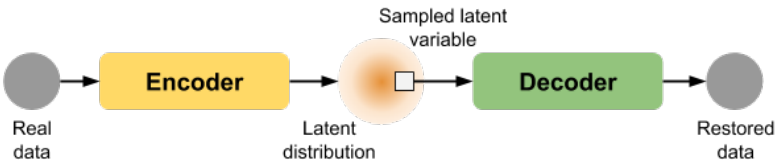


Figure 9: Architecture of a variational autoencoder. The encoder maps the input to a distribution in the latent space instead of to a latent variable as in normal autoencoders. For decoding, a latent variable is sampled from the distribution and, again, the decoder tries to optimally restore the original input.

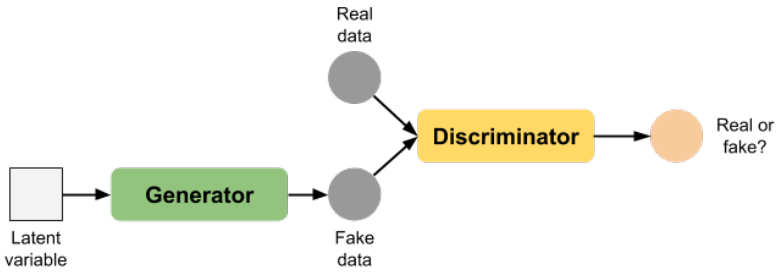


Figure 10: Architecture of a GAN. The generator outputs synthetic data to fool the discriminator, and the discriminator tries to distinguish between original and synthetic data as accurately as possible.

most widespread of the improved approaches is the variational autoencoder (VAE) [84, 85] in which the encoder maps each input to a distribution over the latent space (typically Gaussian, see Figure 9 for a schematic) instead of mapping inputs to specific points deterministically. A point is then sampled according to the distribution and handed over to the decoder to obtain the output. As with standard AEs, the network is optimized to minimize dissimilarity between the input and output. The encoder and decoder can be any type of network; in particular, when RNNs are used, the cell state is a natural choice for the bottleneck / latent space.

### 3.6 Generative Adversarial Networks

The GAN [86] takes a unique approach to generating data. Its basic architecture consists of two networks working against each other: a generator network that transforms a noise vector into output in the original data space and a discriminator that tries to distinguish between the generator's output and the original data (see Figure 10). The generator and the discriminator are trained in an alternating pattern, leading to a minimax game between the opposing parts of the network. The generator is optimized to maximize the error rate of the discriminator, and the discriminator is optimized to minimize the error rate.

Many different architectures can be used in GANs. For example, CNNs have successfully generated of images [87], and RNNs are often applied to sequential data [88]. It is possible to add further covariates to the generator (which are added as input to the discriminator) in addition to the random input vector, allowing for conditional generation [89].

Though successful in capturing continuous distributions GANs present some difficulties when dealing with discrete data such as text [86]. The underlying reason is the non-differentiability of the loss function with respect to discrete random variables and, subsequently, back-propagation through time in the learning phase. Potential solutions include approximating the discrete distribution with a continuous one [90] and modifying the loss function [91]. In [92], the generator is modeled as a stochastic policy in reinforcement learning and overcomes the differentiability issue by adapting the optimization scheme. Section 4.1 discusses GAN-based architectures for generating text in detail.

Another challenge associated with GANs is convergence. The minimax nature of GAN training can cause the training to diverge, or converge to a degenerated optimum. Convergence is particularly noteworthy because it is a characteristic of GANs: a common outcome of GAN training is an optimum in which the generator maps all inputs to only one or a few specific images that the discriminator cannot distinguish from original data. Under our proposed criteria, the outcome would lack diversity (see Section 2.1.4) and be of little practical use. Various modifications to the architecture and especially to the loss functions have been proposed to overcome convergence issues (e.g., [36, 93, 94, 95, 96]). In recent years, GANs have become more popular and are the architecture used in numerous models. For an extensive review of this body of literature see [97, 27, 98, 99, 100].

## 4 Applications

We next review applications of deep generative models to generate synthetic sequential data in a variety of domains. We critically analyze the contributions to this fast-growing literature, evaluate them using our proposed criteria, and demonstrate that the criteria individually are not equally relevant in all domains and are not measured the same way. Each subsection discusses the applications in their focal domain and summarize a few representative contributions in terms of the proposed assessment criteria (see Tables sections 4.1 to 4.5). Additionally, we analyze the architectures of

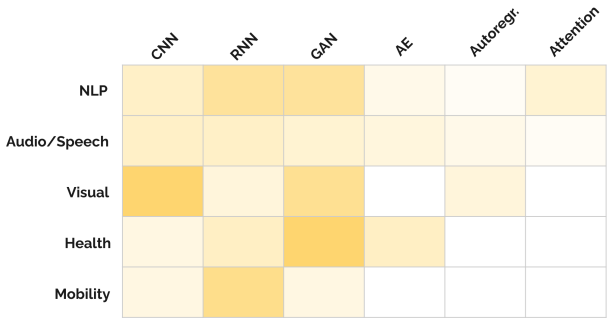


Figure 11: Prevalence of architectures discussed in Section 3 in five domains. A lighter color means that the elements are less prevalent and a darker color indicates a greater prevalence.

the models used in the selected publications. Figure 11 presents an overview of the prevalence of architectural elements used in the reviewed articles in different domains.

### 4.1 Natural Language Processing

| Study | Representativeness | Novelty | Realism | Diversity | Coherence |
|---|---|---|---|---|---|
| Guu et al. [101] | Perplexity | Qual. | Human eval. | Qual. | Human eval. |
| Shen et al. [102] | NLL / perplexity | | Human eval. | Self-BLEU / unique n-grams / 2-gram entropy | Human eval. |
| Fedus et al. [88] | Perplexity | | Human eval. | Unique 2,3,4-grams | Human eval. |

Table 1: Excerpts of studies of generative models for natural language processing and metrics used for evaluation.

NLP is a broad field devoted to computers interacting with human language. Common tasks in NLP include language modeling, text translation, human-machine dialog generation, and natural language generation (NLG). Thanks to widespread adoption of machine learning and deep neural networks in recent years, the research community has made significant progress in accomplishing these tasks. Today, highly capable language models can generate texts that are almost indistinguishable from human-generated text.

Most language data-sets are comprised of text, which can come in many different flavors - news articles, product reviews, medical diagnoses, and music lyrics. However, all text can be represented as a combination of tokens from a discrete vocabulary. The tokens are the most basic components of text, commonly single words complemented with punctuation.

Sentences, paragraphs, and longer texts are then merely sequences of such tokens. But the sequences must obey certain grammatical, semantic, and logical rules. And since sentences are not just loosely strung together, later sentences and words in text can be highly dependent on words that appeared multiple sentences before. For example, a character in a short story that disappears in the beginning can reappear paragraphs later. The rules and contextual dependencies of a text pose significant challenges on language models and on the generation of synthetic text. A model must be capable of capturing the right setting of various linguistic features such as syntax, semantics, pragmatics, and morphology. Otherwise, the resulting text can quickly become incoherent or unrealistic.

Guu et al. proposed a particularly interesting language model in [101]. It was inspired by how humans create complex texts, which rarely arise from scratch in a single pass. Instead, humans rather create initial drafts and revise the drafts incrementally. [101] adopted this idea in their *neural editor* model by sampling a prototype sentence from the training corpus, combining it with a random parameter for editing the sentence, and generating a modified, new sentence. Their edit parameter can lead to changes such as altered wording, shorter or longer sentences, and change from active to passive voice. Architecturally, the model is based on a VAE with an attention-based LSTM encoder and an LSTM decoder. The prototype sentence and the edit parameters are randomly sampled and then used to transform the sentence in a sequence-to-sequence fashion.

A metric commonly used to evaluate the quality of language models is the perplexity [103], which captures how "surprised" a language model is to see the words the original training corpus in terms of probabilities it assigns to each word. Looking at language models as generative, perplexity measures the representativeness of the generative model. [101] evaluated their neural editor using a data-set of restaurant reviews from Yelp and a more-general text data-set. In both cases, they found that the synthetic texts were representative when measured by perplexity. Though they were able to generate novel sentences that were significantly different from the prototype sentences, each synthetic sentence still originated from a single prototype sentence and thus was somewhat close to the prototype.

The edit parameter can be used to perform similar edits on multiple sentences or to smoothly vary the degree to which editing is performed on a single sentence. [101] used these properties to generate a variety of sentences, qualitatively suggesting that generation of diverse data-sets is possible. Individuals deemed the synthetic sentences to be realistic and coherent according to their ratings of overall quality, grammaticality, and plausibility.

Though the neural editor is effective in generating synthetic sentences, the task of creating longer text samples composed of several coherent sentences that are non-repetitive, grammatically correct and non-contradictory remains challenging. Models capable of that task require greater capacity to capture the long-term dependencies in such texts. In [102], Shen et al. proposed such a model. They chose a hierarchical VAE architecture given the inherent hierarchical paragraph structure of longer texts. The encoder network consists of one low-level CNN that maps each sentence to a latent variable and one high-level CNN that maps all the latent variables for each sentence into one latent variable for the entire text input. On the decoding side, two hierarchical LSTM networks operate the other way around at the sentence and on word level. The decoder obtains a latent variable for a text and transforms it via the sentence-level LSTM into latent sentence variables. The sentence-level latent variables are then passed down to the word-level LSTM, which generates the words for the synthetic sentences. By putting all the words together into sentences and the sentences into paragraphs, the model can output longer synthetic paragraphs. Passing the latent variables down the LSTM hierarchy allows the decoder to capture relatively coarse characteristics of text and sentences, such as the topic and sentiment.

Shen et al. [102] evaluated their model using Yelp reviews and abstracts from arXiv papers and found that their multilevel-VAE (ml-VAE) model improved representativeness of the output relative to a flat VAE model (the baseline). They evaluated representativeness by measuring the perplexity of the language model and calculating the corpus-level bilingual evaluation understudy (BLEU) score of the output. The BLEU score was originally developed for in text translation and has proven to be a good metric for measuring translation quality that correlates well with human evaluations. It measures similarities between the generated text and a set of references by comparing their $n$-grams: $n$ consecutive words/tokens in a text. When the set of references is the whole synthetic data-set, the BLEU score is called self-BLEU.

The average BLEU score of the ml-VAE model obtained by comparing generated text to the training corpus also indicated that representativeness was improved relative to the baseline. The authors also reported an acceptable diversity score. Diversity was especially important to them because VAEs used for NLG often suffer from mode collapse. They evaluated diversity by calculating self-BLEU scores, the percentage of unique $n$-grams, and 2-gram entropy of a set of synthetic texts. They further evaluated the coherence and realism of the synthetic text by asking individuals to compare text generated by the baseline model to the ml-VAE synthetic text and choose the one that seemed most "real" to them. Individuals rated the texts' fluency, grammar, and consistency to measure its coherence. These human evaluations also showed that, in terms of realism and coherence, the ml-VAE yielded results that were superior to the results of the baseline model and acceptable when compared to human-generated text.

Likelihood-based models such as VAEs have their critics, who suggest that the models are well suited to optimizing perplexity and representativeness but lack the ability to generate realistic, coherent high-quality samples. Fedus et al. in [88] attempted to generate higher-quality samples using a GAN-based model that incorporated LSTM encoder-decoder networks in the generator and discriminator. To improve overall training, they masked the sentences by blanking words and asking the generator to predict the missing words based on the rest of the sentence. In that case, the networks knew the entire context of the sentence; most other models condition a word solely on the preceding words in the sentence. They found that their hybrid GAN model improved perplexity and thus representativeness relative to a likelihood-based baseline model. Still, they claim that low perplexity alone is not indicative of high-quality synthetic text, their primary focus. Their human evaluations also showed that the hybrid GAN model produced more realistic samples than the baseline model in most cases, distinguishing between the synthetic and human-generated texts seems to be relatively easy for the participants. Since mode collapse is a common issue in GANs, the authors also took a narrow look at the diversity of the synthetic results. They evaluated the percentages of unique 2-, 3-, and 4-grams and found some mode collapse, indicating that the text generated by their model lacked diversity relative to the text generated by the baseline model. In addition the synthetic sentences sometimes lacked coherence because they lost the global context. However, the authors expected to be able to improve coherence by increasing the capacity of the model.

NLP is a heavily researched domain that has produced a wide range of applications. The primary concern of most studies of generative models is generation of representative and realistic synthetic texts with realism implicitly used as a metric for coherence in most cases. Researchers also are concerned about mode collapse in generative models; when their models are prone to that failure, diversity is investigated in detail. However, novelty is rarely addressed and could be of interest primarily in privacy-sensitive cases such as medical patients' chief complaints [104]. Interestingly, for most of our high-level evaluation criteria (Section 2) some metrics have been established for NLP. NLL, BLEU, and perplexity are often used to measure representativeness. Realism and coherence are mostly evaluated together as parts of human evaluation studies with participants choosing between synthetic and human-generated text based on a variety of properties. Finally, to assess the diversity of synthetic results, studies used either self-BLEU or statistics such as the percentage of unique $n$-grams.

### 4.2 Speech and Audio Processing

| Study | Represen-tativeness | Novelty | Realism | Diversity | Coherence |
|---|---|---|---|---|---|
| Van den Oord et al. [40] | | Qual. | Qual. | | Qual. |
| Dieleman et al. [105] | | | Human eval. | | Qual. |
| Mehri et al. [106] | ALL | | | | Qual. |
| Dong et al. [107] | Music-theoretical measures | | Human eval. | | Tonal distance / Human eval. |
| Donahue et al. [39] | | Avg. Eucl. distance to original set | IS / Human eval. | Avg. Eucl. distance to synth. set | |

Table 2: Excerpt of studies of synthetic speech and audio data and metrics used to evaluate the output.

Generation of audio data has a long history. It originated in several quite different domains and relied on completely different theories. Most notable the generation of synthetic music and speech. Both ultimately make data audible by converting it to sound. As different as these origins and the rules used to generate synthetic sound are, both are specific types of digital audio data that eventually yield the same result.

Following the success of deep neural networks in generating content such as images, video,a and text and the availability of vast quantities of audio data, researchers began to apply the techniques to audio-synthesis. The resulting models learned either from raw audio signals or from intermediary representations such as musical scores and linguistic speech parameters. The models can grasp the underlying structure of the data to create realistic-sounding synthetic audio data.

The most general representation of sound is the amplitude of sound waves over time sampled at a constant rate (i.e., raw audio). Consequently, the sound signal is continuous and one-dimensional. Still, because of the high frequencies of natural sounds, the sequences are long and complex. Typical sampling rates are at least 16kHz, resulting in signals with thousands of steps per second.

Models designed to work with raw audio generally are the most adaptable. Unlike models that use intermediary representations, their results do not have to go through one or more conversion steps before becoming audible [44]. The drawback of raw audio is the need high-capacity models that can learn certain rules on their own instead of having to encode the rules in specific representation. For example, to generate realistic speech, models have to learn how intonation affects meaning to generate realistic speech. Speech parameters already encode intonation rules to some extent.

Deep learning models can leverage some aspects of audio data by choosing appropriate representations. But, as previously mentioned, there are drawbacks. Musical scores, for example, require multiple conversion steps to become audible. Additionally, representations can abstract away relevant nuances of music and speech. For example, timing and volume can be important when generating synthetic music, but often cannot be represented accurately in musical scores.

When generating music and speech, use of raw audio signals in generative models is in the minority. Applications such as WaveNet [40] show that raw audio models can succeed in multiple domains by leveraging the flexibility of deep learning models. WaveNet is an autoregressive model that predicts one step of a sequence at a time conditioned on previous steps. Multiple layers of causal convolutions incorporate causality into the network. These are one-dimensional convolutions that depend only on present and past time steps. A key problem of networks involving causal convolutions is that, when the convolutions depend on the present and previous time steps, the network has to be quite deep to capture long-term dependencies. WaveNet [40] overcomes this obstacle by dilating the convolutions in each layer. Therefore, instead of using the output of the preceding time step as input, WaveNet skips multiple time steps (see Section 3.2).

The WaveNet [40] model has been evaluated in numerous experiments. Most important for this review is the unconditional generation of polyphonic single-voice piano music and of speech for a single speaker. WaveNet made a significant leap forward in the ability to generate of synthetic audio data by adopting deep learning models and still serves as a baseline for evaluation of new models. [40] addressed novelty, realism, and coherence of the sample output of WaveNet only qualitatively and did not address representativeness or diversity. Qualitatively, the synthetic music was rated as harmonious and aesthetically pleasing. Their synthetic speech samples consisted of non-existent words that resembled actual words and were spoken with realistic intonations. The authors argue that conditioning on information such as a speaker's ID for speech and genre for music, yields better results. Additionally, because the input size was limited, WaveNet's synthetic outputs lacked long-term coherence and synthetic music samples sometimes changed genre and volume from one second to another.

The structure of raw audio makes generation of long coherent audio signals challenging. The signal at one time step can depend on the values of neighboring time steps and on the values of thousands of preceding time steps. WaveNet lacks this long-term coherence but yields short audio samples of good quality. To overcome this limitation, WaveNet has been incorporated into higher-level architectures (e.g., [105, 106]). Dieleman et al. [105] transformed raw audio signals into a more-abstract, higher-level representation and train WaveNet on the representation.

In their SampleRNN model, Mehri et al. [106] addressed the problem of coherence by hierarchically stacking networks that operated at different timescales. The lowest layer of the SampleRNN is a WaveNet network operating on the raw audio signal. Higher layers operate on coarser timescales by collating multiple time steps of the signal into the state of an RNN. As a result, the higher layers can capture long-term dependencies and pass that information down the network hierarchy, allowing WaveNet to obtain aggregated dependency information from numerous preceding time steps. The SampleRNN was evaluated on speech data, human sounds, and music data. The authors reported that it generated more-representative synthetic audio samples than a simple WaveNet, based on the NLL of the synthetic samples. Also, participants who evaluated the results of SampleRNN in an empirical study perceived the synthetic output more realistic than the output of WaveNet.

Donahue et al. [39] also worked with raw audio but applied an interesting approach. They transferred the DCGAN network [87], a model prominently known for its success in image synthesis, to audio generation. They created two models: WaveGAN for raw audio and SpecGAN for spectrograms of sound data. Both are GAN models with a convolutional generator and discriminator and a structure similar to DCGAN. However, since as raw audio is one-dimensional and images are two-dimensional, the convolutions are flattened. For example, two-dimensional filters sized 5x5 in DCGAN become one-dimensional filters of length 25 in WaveGAN and WaveGAN's output is a raw audio sample of length 16,384 instead of an image of size 128x128. SpecGAN, on the other hand, operates on the two-dimensional spectrograms of raw audio data. The raw audio samples are first transformed into intensity distributions of different frequencies at each timestep, creating spectrograms. SpecGAN then generates synthetic two-dimensional spectrograms that are inverted back to raw audio to obtain audible sound.

With a sampling rate of 16kHz WaveGAN and SpecGAN generate synthetic audio samples that have a duration of about one second. The models are applied to data-sets with similarly short sounds, such as intonations of the numbers zero through nine in speech, short drum and piano sounds, and bird vocalizations. The authors thoroughly evaluated the two models using IS, nearest-neighbor comparisons, and human judgement. Donahue et al. [39] used the IS, which was originally developed to evaluate of synthetic images, to determine the realism and diversity of their synthetic sounds. To evaluate diversity, they measured the mean Euclidean distance between a synthetic sound and its nearest neighbors. Novelty was determined by the mean Euclidean distance between a synthetic sound and nearest neighbors in the original data-set. Additionally, study participants evaluated the quality, diversity, and realism of the synthetic vocalizations of the numbers. The authors report better results in terms of novelty, diversity and realism than achieved using SampleRNN [106] and WaveNet [40].

The limitations associated with using raw audio data in terms of sequence length make use of higher-level representations of sound such as musical scores and the Musical Instrument Digital Interface (MIDI) standard for music beneficial in some scenarios. Higher-level representations can encode important information but abstract away some aspects of raw audio. Less capacity is needed for these models, but the representations cannot be made audible directly. Often, some

interpretation is to musicians or to computer programs. Additionally, abstraction reduces sequence length while usually increasing dimensionality.

Piano rolls are an example of a higher-level representation of music. They were inspired by the rolls used in automated pianos that triggered playing of a note for a certain duration. Similarly, piano roll representations encode whether a note —or multiple notes in polyphonic cases —is played in a particular time step of a song. The duration of the time steps is constant for a single piano roll and across a data-set. The duration is much longer than in raw audio data so piano rolls can encode multiple seconds of melodies using shorter sequences and thus make it easier to capture intra-sequence dependencies. However, piano rolls slightly increase dimensionality because each note in a track is encoded instead of amplitudes of sound waves. There are several other representations used for music, and the literature on deep learning models for generating symbolic music is extensive [46, 108, 49, 41, 107, 109, 110, 44, 111, 112, 113, 14, 114, 115]. It is partially reviewed in a survey by Briot et al. [31].

In [107], Dong et al. described a model designed to generate multi-voice polyphonic rock music called MuseGAN operating on piano rolls. Multi-voice music consists of multiple tracks for the instruments (e.g., piano, guitar, and bass). Each track is represented by a piano roll. The challenge in modelling multi-voice polyphonic piano rolls is capturing the intra-dependencies of notes in a track and the inter-dependencies of notes played in different tracks.

MuseGAN [107] uses the intra- and inter-dependencies of tracks to compose synthetic music, further separating the dependencies into time-dependent and time-independent parts. The network is a GAN that uses a generator partly inspired by generative video models [42, 47, 48] (see Section 4.3 for details on these models). The synthetic music is sampled from the generator by track. Each track is generated from two random numbers representing all tracks and two random numbers representing individual tracks encoding time-dependent and time-independent intra- and inter-track dependencies. The track-generator captures dependencies in time and between notes played using a CNN structure. Similarly, the MuseGAN discriminator is a CNN that judges whether a melody is real or synthetic based on the structure of the notes played in a single track over time and in multiple tracks at the same time.

Dong et al. [107] leveraged symbolic representation to reduce the complexity of the problem and to assess the quality of the generated music samples. To evaluate the representativeness and coherence of the meusic they compared the training data and synthetic data based on music-theoretical measures. For example, they computed the ratio of bars in which no notes were played, the number of pitch classes used in a bar, and the ratio of notes lasting longer than a 32nd note to evaluate representativeness. The model captured drum patterns observed in the training data fairly well, but the synthetic melodies were more fragmented and used a larger number of pitch classes than the original melodies, indicating noise in the synthetic data. The tonal distance [116] between tracks in the generated samples generally showed a strong harmonic relation, indicating strong coherence. In addition to these objective measures, the authors evaluated the synthetic samples' *harmonicity*, *rhythmicity*, *musical structure*, and *coherence* based on responses by study participants, who also gave the samples *overall ratings* that measured coherence and realism as defined in our proposed framework. The study participants rated the samples as $2.3$ to $3.5$ on a $1$–$5$ scale; they did not compare the samples to baseline samples from other models or to the original music.

Speech also can be generated using representations. One of the most studied paradigms is statistical parametric speech synthesis (SPSS) [117], which uses linguistic features of speech such as phonemes, cadence, and word frequency to synthesize spoken words. Considerable research has been conducted on SPSS, but unconditional generation of synthetic speech data-sets is uncommon. Common tasks are text-to-speech, voice conversion, and vocoding (making speech parameters audible). In all three, cases speech is being generated from an input (text, speech fragments, speech parameters). Though these tasks fall outside the scope of our literature review it is important to note that deep learning based models for speech data are emerging (see [118, 119, 120] for example).

Evaluation of synthetic audio data poses a challenge that cannot adequately be addressed in general: the significance of our proposed criteria and validity of metrics used to measure the criteria vary with the type of audio (e.g., speech versus music). For example, rhythms and harmony are highly relevant for music but only somewhat relevant for speech. Reasonable evaluations are often based on domain-knowledge. In the case of music, the relevant domain is music theory, for which metrics such as the ratio of pauses, fragmentation of a sample, and the tonal distance, as used in [107], are reasonable. For a review of objective metrics for evaluating synthetic music, see [121]. For all kinds of audio and for music and speech in particular, subjective evaluations of realism and coherence by humans are a significant part of evaluations.

### 4.3 Visual Data Processing

Today, thanks to the prevalence of smartphones, images and videos are produced and consumed en masse. Access to such a vast amount of data has led to dramatic advances in processing and classification of existing images and in models to generate synthetic ones (see for example [87, 1]). Since videos are merely sequences of images, the ability

| Study | Representativeness | Novelty | Realism | Diversity | Coherence |
|---|---|---|---|---|---|
| Vondrick et al. [42] | Qual. | | Human eval. | | |
| Saito et al. [47] | | | IS | IS | |
| Tulyakov et al. [48] | | Qual. | IS / Human eval. | IS | ACD |

Table 3: Excerpt of models for synthetic videos and metrics used to evaluate them.

to generate synthetic videos also has advanced. The ongoing challenge is capturing a smooth dynamic motion in the transitions between images.

Models based on CNNs (Section 3.2) and GANs (Section 3.6) have been highly successful in generating images. Consequently, many successful generative models for synthetic videos have been based on them [42, 47, 48]. The primary challenge in designing such models is incorporation of the temporal dimension with videos' two spatial dimensions of the video.

The VGAN model proposed in [42] tackles this challenge by decomposing the dynamic foreground from the static background, reducing the complexity of the problem. The dynamic foreground is captured by a three-dimensional spatio-temporal CNN, and the static background can be captured by a two-dimensional spatial CNN. Both CNNs are incorporated into the generator of a GAN that is then optimized against a three-dimensional spatio-temporal discriminator that judges the realism of the scene and the motion.

The VGAN model has been applied to small short videos of 64x64 pixels with 32 frames and duration of around one second from Flickr in different categories collected such as beaches, golf courses, and train stations. The authors [42] assessed the representativeness of the resulting synthetic videos qualitatively and reported generally correct motion patterns for scenes in the various categories. For example, synthetic videos of beaches contained crashing waves and synthetic videos of trains contained train tracks and train cars with windows moving by quickly. The generated scenes were sharp overall, but individual objects such as people in the synthetic beach scenes tended to lack resolution. The realism of the resulting videos was evaluated by participants in an empirical study who were asked to view the synthetic and original videos and choose which seemed most realistic. Though the participants overwhelmingly chose the original videos, the synthetic scenes were chosen in 18% of the comparisons.

The VGAN architecture [42] is optimized for videos with static backgrounds. Saito et al. [47] relaxed this restriction in their TGAN model by decoupling the temporal dimension from the spatial dimensions. First, a one-dimensional temporal generator produces a sequence of temporal codes that are then mapped one-by-one to an image by a two-dimensional image generator. The discriminator, a three-dimensional spatio-temporal CNN, then distinguishes real videos from synthetic ones. According to the IS, the synthetic videos generated by TGAN are more diverse and realistic than those generated by VGAN.

Tulyakov et al. [48] argued that the straightforward decomposition of a video into temporal and spatial dimensions, as done in TGAN, unnecessarily increases the complexity of the problem by ignoring similar motion patterns. In [48], they proposed a decomposition of the content of a video and the motion therein, which they incorporated into a generative model called MoCoGAN. Consider, for example, various facial expressions presented by a person in a video. In such a video, the person's face is the content and performance of an expression is the motion. This disentanglement allows the model to generate videos with the same content but different motions and vice versa —that is, videos of a person performing different facial expressions.

MoCoGAN incorporates this decomposition in the latent space of the generator. The input to the generator is split into a content variable and a sequence of motion codes. An RNN generates the motion codes and connects them to subsequent codes to ensure a coherent motion. Then, given the fixed content variable for all frames and a motion code, each frame in the video is synthesized by a two-dimensional CNN image generator. Similarly, the discriminator judges the realism of the content and motion of a video separately using a two-dimensional CNN for the content and a three-dimensional spatio-temporal CNN for the motion.

The authors evaluated the MoCoGAN's performance synthesizing small short videos of various scenes, including tai-chi movements and facial expressions. They qualitatively assessed the ability of the model to decompose content from motion by fixing a person as the content and generating videos of that person performing different motions. The results demonstrated MoCoGAN's ability to generate novel content by adjusting the input variables of the generator.

They found that the synthetic videos generated by MoCoGAN were more diverse and realistic than synthetic videos generated by VGAN [42] and TGAN [47] based on the IS. Additionally, participants in an empirical study viewed the videos generated by MoCoGAN as more realistic than videos generated using VGAN [42] and TGAN [47]. Tulyakov et al. [48] also quantitatively evaluated the coherence of synthetic videos of facial expressions using the classifier-based average content distance (ACD), which quantifies the difference between two frames in a video in terms of content. OpenFace [122] is applied to each frame of a video presenting a facial expression to extract facial features that identify the person. Small differences (distances) in the features between frames indicate that the same person is displayed throughout the video and, therefore, a small ACD. The MoCoGAN obtained higher coherence scores than the VGAN [42] and TGAN [47] videos.

When generating synthetic videos, many concepts from image generation carry over. We see this in the prevalence of CNN and GAN models and in the metrics used to evaluate synthetic videos. Specifically, the IS is often used to measure realism and diversity of synthetic videos and [48] uses ACD to measure coherence; both rely on image classifiers. When evaluating reaslism, human studies are heavily used in addition to IS and ACD. Human evaluations of realism also capture coherence to some extent. The representativeness and novelty of synthetic videos are rarely evaluated explicitly. Altogether, the results so far are promising for synthetic videos that are short and relatively low resolution. The large number of dimensions associated with high-quality videos combined with the large number of frames needed even for short videos continue to thwart efforts to synthesize more complex videos.

### 4.4 Healthcare

| Study | Represen-tativeness | Novelty | Realism | Diversity | Coherence |
|---|---|---|---|---|---|
| Esteban et al. [123] | MMD/TSTR | NN distance | TRTS | Qual. | |
| Choi et al. [45] | Qual. | Disclosure risk | Human eval. | Qual. | Qual. |
| Baowaly et al. [5] | K–S test/ dim.-wise stats. | Qual. | ML predictions/ ARM | Qual. | ARM |

Table 4: Excerpt of models to generate synthetic medical data and metrics used to evaluate them.

Generative models for synthetic medical data have gained attention in recent years. The sensitivity of medical data and strict access restrictions make sharing of original medical data from patients extremely challenging [43]. A promising solution is to use generated synthetic data instead. Synthetic medical data can be shared and published for secondary analyses since the privacy of patients is guaranteed.

Data from intensive care units (ICUs), where patients with severe and life-threatening conditions first receive treatment, are especially valuable for clinical analysis [124]. The data can include real-valued monitoring information, such as measured oxygen saturation, heart rate, and respiratory rate.

Esteban et al. [123] generated such synthetic medical data based on information collected from the first four hours of patients' stays in an ICU. They employed an LSTM as the generator in a GAN and another LSTM as the discriminator of real and synthetic data sequences. They evaluated representativeness of the generated data using MMD and by training a classifier model on the synthetic data-set and testing it on a real holdout data-set (train on synthetic, test on real (TSTR)). They evaluated realism by training a classifier model on the real data-set and testing it on the synthetic data-set (train on real, test on synthetic (TRTS)). In both cases, the classifier models achieved results comparable to models trained and tested solely on original data.

Novelty is especially important in privacy contexts; that is, it must be impossible to reconstruct the original data-points from the synthetic ones. Overall, Esteban et al. [123] found that the synthetic data-points were not close to original single data-points based on the evaluation of the distances between the synthetic data-points and their real nearest neighbors. Their qualitative exploration of the latent space —conducted by interpolating between generated points —also showed that the model yielded diverse results. To account for the importance of privacy, they adapted the training of the original model to incorporate differential privacy [125, 126]. Under the stricter privacy conditions, they reported that the synthetic data were highly representative and slightly less realistic.

The real-valued time-series data used by Esteban et al. [123] are important in healthcare but are one of many types of electronic health records (EHR). EHR data has been the main focus of recent studies [127] and turns out to be quite diverse. EHRs include patients' demographic information, diagnoses, laboratory test results, medication history, clinical notes, and medical images, and other medical records [128] and disclose discrete-valued codes for diagnoses, medications, and procedures.

Choi et al. [45] studied synthetic sequences of discrete-valued multi-label EHR data containing information on diagnoses and treatments. The sequences in the data were long and high-dimensional, thus presenting significant challenges for generation of synthetic data. The authors addressed these challenges by combining an AE and a GAN in their generative model, medGAN. The AE was used to reduce the complexity of the output data of the generator, which learns salient features of the samples by projecting them to a lower dimensional space and then projecting them back to the original space [129, 130]. Thus, medGAN generates synthetic data in the lower dimensional space. Then, the pre-trained decoder converts the generated output to synthetic EHR data in the original space.

The authors evaluated medGAN and found that it outperformed several generative models, including random noise, independent sampling [45], stacked restricted Boltzmann machines [131] and VAEs (see Section 3.5). Representativeness and diversity are only evaluated qualitatively, but the authors argued that significant improvements were accomplished by applying the minibatch averaging method [45] to reduce overfitting and mode collapse. Novelty was evaluated by conducting two privacy risk evaluations. One measured the risk of disclosure of personally identifiable information and the other measured the risk of disclosure of personal sensitive medical data. The evaluations determined that medGAN can generate novel private synthetic data that reveal little information to potential attackers rather than simply reproducing the training samples. Overall, medGan's synthetic data were reported to be realistic, but qualitative evaluation by a single doctor is not entirely convincing.

Since introduction of medGAN, other researchers have extended it in different directions. Two that have outperformed medGAN in all experiments were proposed by Baowaly et al. [5]. The medWGAN model combines medGAN with the Wasserstein GAN model, which uses a gradient penalty [93, 96] to minimize divergences in Wasserstein distances. The medBGAN (medical boundary-seeking GAN) model trains the generator to obtain a distribution of samples located on the decision boundary of the discriminator. To evaluate the models' representativeness, the authors conducted the Kolmogorov–Smirnov (K-S) test and compare the dimension-wise probabilities and averages of the real and synthetic data. Realism was evaluated by comparing predictions made by machine learning models for the real and synthetic data. Association rule mining (ARM) is often used to identify associations and patterns in clinical concepts in EHR data [132] and was used by [5] to evaluate realism and coherence. Another extension of medGAN for generating real-valued time-series data, has been proposed by Yahi et al. [133].

Medical text and images also have attracted attention. Medical text consists of clinical notes and patients' chief complaints, which share characteristics of other types of text data (see Section 4.1) but typically are short and are composed of a limited number of words from medical vocabularies. Lee [104] applied an encoder-decoder model to generate synthetic natural-language chief complaints using EHR data from around 5.5 million records of emergency department visits. Guan et al. [134] proposed a GAN model to generate Chinese EHR text data. Both models use demographic and disease features as inputs and generate corresponding EHR text data. However, they are conditional models that fall outside the scope of this survey.

In healthcare, synthetic EHR data is primarily used to protect patients' privacy while enabling data sharing and secondary data analyses. Thus, most studies in the field are concerned mainly novelty, representativeness, and realism. Novelty is particularly important to protecting privacy and, thus, is often evaluated using privacy tests. Tests for representativesness and realism in EHRs are not necessarily domain-specific; TSTR and TRTS have most often been used to evaluate those criteria.

### 4.5 Mobility

Everyday, massive quantities of data on human mobility are collected. Mobile devices such as smartphones are equipped with GPS functionality and transportation systems (car sharing, logistics, public transports) usually incorporate automatic tracking. Mobility data are used in a wide range of tasks, including urban traffic predictions [137], shared mobility services [138], marketing services [16], and transportation of people and goods [139]. However, the risk of re-identification of individuals makes sharing of such data highly sensitive. The relevance of this risk has been demonstrated even for aggregated mobility data [140]. Synthetic mobility trajectories do not present this risk and thus enable sharing, by either obfuscating the original path data or generating completely synthetic trajectories that cannot be related to individuals.

Ouyang et al. [135] studied generation of synthetic realistic human location trajectories for privacy-sensitive secondary data analyses. Usually, mobility trajectories are represented as sequences of continuous coordinates $(x, y)$ consisting of

| Study | Represen-tativeness | Novelty | Realism | Diversity | Coherence |
|---|---|---|---|---|---|
| Ouyang et al. [135] | absol. semantics, marg. distributions | | | | rel. semantics |
| Kulkarni et al. [6] | absol. semantics, MMD | location hiding | | | temporal dep. decay |
| Lin et al. [136] | counting stat. | | | | |

Table 5: Excerpt of models for generating mobility data and applied metrics.

a longitudinal and latitudinal component over time $t$. Ouyang et al. converted this time-major representation into a location-major representation in the form of maps corresponding to times of stays at each coordinate $(x, y)$. The maps were then fed into a GAN consisting of a deconvolutional generator and a convolutional discriminator.

The authors evaluate the model results primarily in terms of representativeness and coherence. Representativeness was evaluated by comparing geographical statistics describing the real data with the same statistics for the synthetic data. They compare the marginal probabilities of visiting a certain location at a certain time and of remaining there for a certain duration using Jensen-Shannon divergence (JSD).

The so called *semantics* of the trajectories play a key role in producing representativeness and coherence. The semantics give a trajectory intrinsic meaning, which can be difficult for generative models to capture. Consider, for example, the path "home-bus-work-bus-home". It intuitively makes sense whereas "airport-home-work-train" does not make sense and semantically is unlikely to be true. Ouyang et al. further distinguished between absolute and relative semantics. Absolute semantics captures the meaning of each location in a trajectory; relative semantics capture the meaning of a location in a trajectory relative to other visited locations in the trajectory. To evaluate representativeness, the authors compared the absolute semantics of the real and synthetic data at the population level. Likewise, they measured coherence using a comparison of the relative semantics measured by the pair-wise semantic distance which was originally introduced by Bindschaedler and Shokri [141]. This metric accounts for trajectories of people who can live in geographically different locations but still share semantic patterns. Their results showed that the GAN-based approach preserved both the statistical characteristics of the original data and their relative semantics.

Ouyang et al. [135] did not conduct any privacy tests and limited the evaluation to one GAN-based model. Kulkarni et al. in [6] extended their study by testing the performance of seven generative models that used different architectures and conducting privacy tests to measure the novelty of the results. They compared deep generative models based on GANs, LSTMs, and other variations of RNNs with each other and with a statistical model, Copulas. Interestingly, Copulas and the GANs performed best in terms of representativeness, which was evaluated by comparing geographical statistics and absolute semantics (similar to [135]) and by measuring MMD. The RNNs and Copulas generated the most coherent synthetic trajectories. The long-range temporal dependencies throughout the generated trajectories, which measure coherence, decayed most slowly.

Interestingly, Kulkarni et al. measured the novelty of the synthetic data by conducting two specific privacy tests. They applied a *location-sequence attack*, which determines the level of accuracy to which trajectories in the original data can be reconstructed, and a *membership interference attack*, which measures the accuracy of an inference that an individual contributed to a specific trajectory. In both tests, the RNN and GAN models outperformed the other models by a considerable margin.

The synthetically generated data in [135] and [6] were intended to be used in privacy-sensitive secondary data analyses. This is an important use, but the value of synthetic mobility data extends far beyond that. In [136] Lin et al. used labeled cellular geo-location data collected from mobile devices to generate synthetic mobility data for traffic volume simulations. Actual high-quality data on traffic volumes are difficult to collect. The simulations were applied to a super-district in the San Francisco Bay Area in California and were used to provide decision support for several transportation projects designed to improve urban transportation planning. The authors employed an LSTM model and evaluated its representativeness by comparing the vehicle traffic counts and public transit boarding and alighting counts of the simulated results and the actual counts. They argue that transportation policy-makers and planners can benefit from using synthetic location data to improve their understanding of urban mobility.

The literature on models for generating mobility data is not vast, and the quantitative approaches used to validate such models vary greatly. In reviewing different applications to mobility data, we observed that representativeness was

particularly important in all of the studies. Consequently, the studies provide reliable metrics for representativeness , such as MMD, JSD, absolute semantics, and count variables. Coherence seems to be important in many cases but is evaluated in various ways, including relative semantics of the trajectories and observations of decays of temporal dependencies throughout the trajectory. Privacy, of course, is a significant issue. [6] especially stands out its consideration of privacy as the authors conducted robust privacy tests on their synthetic data. We also find that all of the reviewed studies related to mobility presented a strong use-case for the generative models.

## 5 Summary & Conclusions

Synthetic data allows governments, businesses and researchers to easily access and share sensitive data without the risk of violating privacy regulations. The importance of having access to highly sensitive data was highlighted once again through the COVID pandemic, where governments and researchers rely on high quality sensitive medical data. Furthermore, the decomratizing effect of accessible synthetic data mitigates the power of large data aggregators, such as Google and Facebook, and reduces limitations of real world data-sets, such as inherent biases, insufficient quantities, and class imbalance. Deep generative models, with their ability to capture complex data and their relationships, has boosted progress in synthetic data generation significantly.

This article discusses deep generative models for synthetic data and introduces a set of high-level evaluation criteria for a data-driven assessment of the quality of generated data. We examine their use and applicability to synthetic sequential data in the fields of natural language, speech, audio and visual data processing as well as healthcare and mobility. The proposed evaluation framework allows for clear and easy communication of the requirements posed on synthetic data in different domains and use-cases. We find that synthetic texts in NLP applications are primarily evaluated for representativeness and realism. Synthetic music, speech and video data mostly need to be realistic and coherent. Studies in healthcare are mostly concerned with generating private synthetic EHRs that still allow for secondary data analysis and, thus, assess the data's representativeness, novelty and realism. Synthetic mobility trajectories are generated for similar purposes with an additional focus on their coherence. However, not all mobility studies examine the synthetic data's novelty, potentially leading to privacy risks when sharing such data.

We also find that the nature of metrics used to evaluate the criteria can vary significantly. Some studies evaluate criteria only qualitatively, for example, by looking at synthetic text samples or listening to synthetic music samples. In most cases only the individual-level criteria (i.e., novelty, realism, and coherence) are evaluated in this subjective way, but sometimes also representativeness and diversity are. Other studies rely on human evaluations by laypeople or experts to judge realism and coherence of synthetic data. Human evaluations often also contain subjectivity either by the designer or the participants of the study. Thus, the most objective measures are formal computational metrics. Such metrics are primarily used to evaluate representativeness (e.g., by MMD or NLL) and diversity (e.g., self-BLEU or distances) of synthetic data. In many cases novelty is not evaluated at all and coherence is assessed as part of evaluating the data's realism.

Our review highlights that the generative architectures discussed in Section 3 are used in a variety of applications and, in particular, GANs receive a lot of attention. Most often the architectural elements are used in conjunction with each other. In many cases, at the core of the networks RNNs or CNNs are involved to ensure causally coherent generation of synthetic sequential data. Autoregressive elements and attention mechanisms are also applied to some use-cases.

For the future, our proposed evaluation framework for unconditionally generated synthetic data has potential to be extended for the evaluation of conditionally generated data. That kind of data is always generated within a given context, such as categories of videos or genres of songs. An evaluation framework for conditionally generated synthetic data has to account for that context. We expect conditionally generated synthetic data need robustness within a context and it's variability to be more nuanced depending on the context.

With more jurisdictions passing privacy laws, in the future, we expect synthetic data to gain more attention. We expect more advanced and more objective metrics that allow a better and more objective assessment of synthetic data quality, in particular on the individual level. The development of the IS, used for synthetic images and videos, and other metrics that correlate well with human judgement of realism point in that direction. In depth research of objective metrics allows systematic assessment of synthetic data quality with more robustness and less subjectivity in it. Meanwhile, we expect a continuation of the coexistence and combination of quality assessment based on expert judgement and formal computational metrics.

Another potentially interesting area worth further exploring is to complement a purely data-driven approach to assess the quality of synthetic data with a decision-oriented view. Credible decisions made on the basis of data can require certain properties of the data. For example, biased data with underrepresented minority groups can be a weak basis for decisions influencing all individuals, including the minority group. Other decisions can be sensitive to recent events

in the data. The decisions made during the COVID pandemic, for example, are highly sensitive to the recency of the data. A decision-oriented evaluation approach could help improving decision making (or avoiding weak decisions) by contrasting the decisions derived from synthetic data-scenarios with those based on the original, real-life data. Recent research into fairness and debiasing using synthetic data are promising starting points towards this direction.

## Acknowledgements

This paper is supported by the "ICT of the Future" funding programme of the Austrian Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology.

## References

[1] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5908–5916, 2017.

[2] Lakshmi Kurup, Meera Narvekar, Rahil Sarvaiya, and Aditya Shah. Evolution of neural text generation: Comparative analysis. In *Advances in Computer, Communication and Computational Sciences*, pages 795–804. Springer, 2021.

[3] Jean-Pierre Briot. From artificial neural networks to deep learning for music generation: history, concepts and trends. *Neural Computing and Applications*, 33(1):39–65, 2021.

[4] S. Norgaard, R. Saeedi, K. Sasani, and A. H. Gebremedhin. Synthetic sensor data generation for health applications: A supervised deep learning approach. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1164–1167, 2018.

[5] Mrinal Kanti Baowaly, Chia-Ching Lin, Chao-Lin Liu, and Kuan-Ta Chen. Synthesizing electronic health records using improved generative adversarial networks. *Journal of the American Medical Informatics Association*, 26(3):228–241, 2019.

[6] Vaibhav Kulkarni, Natasa Tagasovska, Thibault Vatter, and Benoît Garbinato. Generative models for simulating mobility trajectories. *CoRR*, abs/1811.12801, 2018.

[7] Magnus Wiese, Robert Knobloch, Ralf Korn, and Peter Kretschmer. Quant gans: deep generation of financial time series. *Quantitative Finance*, 0(0):1–22, 2020.

[8] Maayan Frid-Adar, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Synthetic data augmentation using gan for improved liver lesion classification. In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pages 289–293. IEEE, 2018.

[9] Yuanyuan Chen, Yisheng Lv, and Fei-Yue Wang. Traffic flow imputation using parallel data and generative adversarial networks. *IEEE Transactions on Intelligent Transportation Systems*, 21(4):1624–1630, 2019.

[10] Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. Fairgan: Fairness-aware generative adversarial networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 570–575. IEEE, 2018.

[11] Andrew Yale, Saloni Dash, Ritik Dutta, Isabelle Guyon, Adrien Pavao, and Kristin Bennett. Privacy preserving synthetic health data. In *ESANN 2019-European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2019.

[12] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[13] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[14] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer, 2018.

[15] Y. Shavitt and N. Zilberman. A geolocation databases study. *IEEE Journal on Selected Areas in Communications*, 29(10):2044–2056, 2011.

[16] Sam K. Hui, Peter S. Fader, and Eric T. Bradlow. Path data in marketing: An integrative framework and prospectus for model building. *Marketing Science*, 28(2):320–335, 2009.

[17] A. Murat Tekalp. *Digital Video Processing*. Prentice Hall Press, USA, 2nd edition, 2015.

[18] B. Zhang, G. Kreitz, M. Isaksson, J. Ubillos, G. Urdaneta, J. A. Pouwelse, and D. Epema. Understanding user behavior in spotify. In *2013 Proceedings IEEE INFOCOM*, pages 220–224, April 2013.

[19] Lei Huang, Bowen Ding, Yuedong Xu, and Yipeng Zhou. Analysis of user behavior in a large-scale vod system. In *Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and Video*, NOSSDAV'17, pages 49–54, New York, NY, USA, 2017. ACM.

[20] Randolph E. Bucklin and Catarina Sismeiro. Click Here for Internet Insight: Advances in Clickstream Data Analysis in Marketing. *Journal of Interactive Marketing*, 23(1):35–48, 2009.

[21] Xiao-Xi Fan, Kam-Pui Chow, and Fei Xu. Web user profiling based on browsing behavior analysis. In Gilbert Peterson and Sujeet Shenoi, editors, *Advances in Digital Forensics X*, pages 57–71, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[22] Behrooz Noori. An analysis of mobile banking user behavior using customer segmentation. In *International Journal of Global Business*, volume 8, pages 55–64, 2015.

[23] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and S. S. Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM Comput. Surv.*, 51(5), September 2018.

[24] C. G. Turhan and H. S. Bilge. Recent trends in deep generative models: a review. In *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, pages 574–579, 2018.

[25] A. Oussidi and A. Elhassouny. Deep generative models: Survey. In *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)*, pages 1–8, 2018.

[26] Harshvardhan GM, Mahendra Kumar Gourisaria, Manjusha Pandey, and Siddharth Swarup Rautaray. A comprehensive survey and analysis of generative models in machine learning. *Computer Science Review*, 38:100285, 2020.

[27] Yongjun Hong, Uiwon Hwang, Jaeyoon Yoo, and Sungroh Yoon. How generative adversarial networks and their variants work: An overview. *ACM Comput. Surv.*, 52(1), February 2019.

[28] Ivan Kobyzev, Simon J. D. Prince, and Marcus A. Brubaker. Normalizing flows: An introduction and review of current methods, 2019.

[29] Peter B. Jørgensen, Mikkel N. Schmidt, and Ole Winther. Deep generative models for molecular science. *Molecular Informatics*, 37(1-2):1700133, 2018.

[30] Xiaojie Guo and Liang Zhao. A systematic survey on deep generative models for graph generation, 2020.

[31] Jean-Pierre Briot, Gaëtan Hadjeres, and François-David Pachet. *Deep Learning Techniques for Music Generation – A Survey*. 2017.

[32] Touseef Iqbal and Shaima Qureshi. The survey: Text generation models in deep learning. *Journal of King Saud University - Computer and Information Sciences*, 2020.

[33] Lucas Theis, Aaron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. In *International Conference on Learning Representations*, Apr 2016.

[34] Ali Borji. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179:41 – 65, 2019.

[35] Ali Borji. Pros and cons of gan evaluation measures: New developments, 2021.

[36] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 2234–2242, Red Hook, NY, USA, 2016. Curran Associates Inc.

[37] Michael Platzer and Thomas Reutterer. Holdout-based empirical assessment of mixed-type synthetic data. *Frontiers in Big Data*, 4:43, 2021.

[38] Zhengping Che, Yu Cheng, Shuangfei Zhai, Zhaonan Sun, and Yan Liu. Boosting deep learning risk prediction with generative adversarial networks for electronic health records. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2017-Novem:787–792, 2017.

[39] Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. In *International Conference on Learning Representations*, 2019.

[40] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio, 2016. arxiv:1609.03499.

[41] Olof Mogren. C-RNN-GAN: Continuous recurrent neural networks with adversarial training. (Nips), 2016.

[42] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 613–621, Red Hook, NY, USA, 2016. Curran Associates Inc.

[43] Brett K. Beaulieu-Jones, Zhiwei Steven Wu, Chris Williams, Ran Lee, Sanjeev P. Bhavnani, James Brian Byrd, and Casey S. Greene. Privacy-Preserving Generative Deep Neural Networks Support Clinical Data Sharing. *Circulation. Cardiovascular quality and outcomes*, 12(7):e005122, 7 2019.

[44] Sageev Oore, Ian Simon, Sander Dieleman, Douglas Eck, and Karen Simonyan. This time with feeling: learning expressive musical performance. *Neural Computing and Applications*, 32(4):955–967, 2020.

[45] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. Generating Multi-label Discrete Patient Records using Generative Adversarial Networks. 3 2017.

[46] Mason Bretan, Gil Weinberg, and Larry Heck. A unit selection methodology for music generation using deep neural networks. In *Proceedings of the 8th International Conference on Computational Creativity (ICCC 2017)*, pages 72–79, 2017.

[47] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2849–2858, 2017.

[48] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing motion and content for video generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1526–1535, 2018.

[49] Natasha Jaques, Shixiang Gu, Richard E. Turner, and Douglas Eck. Tuning recurrent neural networks with reinforcement learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017.

[50] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, oct 1986.

[51] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[52] Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural Computation*, 12:2451–2471, 1999.

[53] Alex Graves, Navdeep Jaitly, and Abdel rahman Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *ASRU*, 2013.

[54] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016.

[55] Martin Sundermeyer, Hermann Ney, and Ralf Schlüter. From feedforward to recurrent lstm neural networks for language modeling. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 23(3):517–529, March 2015.

[56] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

[57] Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. Massive exploration of neural machine translation architectures. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1442–1451, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[58] Wei Xu, Alan Ritter, Tim Baldwin, and Afshin Rahimi, editors. *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[59] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202, 1980.

[60] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, May 2017.

[61] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[62] S. Lawrence, C. L. Giles, Ah Chung Tsoi, and A. D. Back. Face recognition: a convolutional neural-network approach. *IEEE Transactions on Neural Networks*, 8(1):98–113, 1997.

[63] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[64] Colin Lea, René Vidal, Austin Reiter, and Gregory D. Hager. Temporal convolutional networks: A unified approach to action segmentation. In Gang Hua and Hervé Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, pages 47–54, Cham, 2016. Springer International Publishing.

[65] M. Holschneider, R. Kronland-Martinet, J. Morlet, and Ph. Tchamitchian. A real-time algorithm for signal analysis with the help of the wavelet transform. In Jean-Michel Combes, Alexander Grossmann, and Philippe Tchamitchian, editors, *Wavelets*, pages 286–297, Berlin, Heidelberg, 1990. Springer Berlin Heidelberg.

[66] Jining Yan, Lin Mu, Lizhe Wang, Rajiv Ranjan, and Albert Y Zomaya. Temporal convolutional networks for the advance prediction of enso. *Scientific Reports*, 10(1):1–15, 2020.

[67] Rui Dai, Shenkun Xu, Qian Gu, Chenguang Ji, and Kaikui Liu. Hybrid spatio-temporal graph convolutional network: Improving traffic prediction with navigation data. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3074–3082, New York, NY, USA, 2020. Association for Computing Machinery.

[68] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2267–2273. AAAI Press, 2015.

[69] Yoon Kim. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882, 2014.

[70] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson. Cnn architectures for large-scale audio classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 131–135, 2017.

[71] Matthew D. Zeiler, Dilip Krishnan, Graham W. Taylor, and Rob Fergus. Deconvolutional networks. In *In CVPR*, 2010.

[72] Asifullah Khan, Anabia Sohail, Umme Zahoora, and Aqsa Saeed Qureshi. A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53(8):5455–5516, 2020.

[73] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

[74] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

[75] Han Zhang, Ian J. Goodfellow, Dimitris N. Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv:1805.08318*, 2018.

[76] Sneha Chaudhari, Varun Mithal, Gungor Polatkan, and Rohan Ramanath. An attentive survey of attention models, 2020.

[77] Yoshua Bengio and Samy Bengio. Modeling high-dimensional discrete data with multi-layer neural networks. In *Advances in Neural Information Processing Systems*, pages 400–406, 2000.

[78] Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, and Hugo Larochelle. Neural autoregressive distribution estimation. *Journal of Machine Learning Research*, 17(205):1–37, 2016.

[79] Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *CoRR*, abs/1601.06759, 2016.

[80] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1096–1103, New York, NY, USA, 2008. Association for Computing Machinery.

[81] Alireza Makhzani and Brendan Frey. k-sparse autoencoders, 2014.

[82] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. Adversarial autoencoders. In *International Conference on Learning Representations*, 2016.

[83] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *International Conference on Machine Learning (ICML)*, 2017.

[84] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[85] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR.

[86] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[87] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[88] William Fedus, Ian Goodfellow, and Andrew M. Dai. MaskGAN: Better Text Generation via Filling in the_____. 1 2018.

[89] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014.

[90] Matt J. Kusner and José Miguel Hernández-Lobato. Gans for sequences of discrete elements with the gumbel-softmax distribution, 2016.

[91] Tong Che, Yanran Li, Ruixiang Zhang, R Devon Hjelm, Wenjie Li, Yangqiu Song, and Yoshua Bengio. Maximum-likelihood augmented discrete generative adversarial networks, 2017.

[92] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 2852–2858. AAAI Press, 2017.

[93] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.

[94] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[95] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of gans, 2017.

[96] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5767–5777. Curran Associates, Inc., 2017.

[97] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumara, Biswa Sengupta, and Anil A. Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65, 2018.

[98] Zhaoqing Pan, Weijie Yu, Xiaokai Yi, Asifullah Khan, Feng Yuan, and Yuhui Zheng. Recent progress on generative adversarial networks (gans): A survey. *IEEE Access*, 7:36322–36333, 2019.

[99] Jie Gui, Zhenan Sun, Yonggang Wen, Dacheng Tao, and Jieping Ye. A review on generative adversarial networks: Algorithms, theory, and applications, 2020.

[100] Zhengwei Wang, Qi She, and Tomas E. Ward. Generative adversarial networks in computer vision: A survey and taxonomy, 2019.

[101] Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics*, 6:437–450, 2018.

[102] Dinghan Shen, Asli Celikyilmaz, Yizhe Zhang, Liqun Chen, Xin Wang, Jianfeng Gao, and Lawrence Carin. Towards generating long and coherent text with multi-level latent variable models. *arXiv preprint arXiv:1902.00154*, 2019.

[103] Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63, 1977.

[104] Scott Lee. Natural language generation for electronic health records. *npj Digital Medicine*, 1:63, 12 2018.

[105] Sander Dieleman, Aaron van den Oord, and Karen Simonyan. The challenge of realistic music generation: modelling raw audio at scale. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7989–7999. Curran Associates, Inc., 2018.

[106] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio. Samplernn: An unconditional end-to-end neural audio generation model. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, pages 1–11, 2017.

[107] Hao Wen Dong, Wen Yi Hsiao, Li Chia Yang, and Yi Hsuan Yang. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pages 34–41, 2018.

[108] Bob Sturm, João Felipe Santos, Oded Ben-Tal, and Iryna Korshunova. Music transcription modelling and composition using deep learning. 2016.

[109] Gaëtan Hadjeres, François Pachet, and Frank Nielsen. Deepbach: A steerable model for bach chorales generation. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1362–1371. JMLR.org, 2017.

[110] Daniel D Johnson. Generating polyphonic music using tied parallel networks. In *International conference on evolutionary and biologically inspired music and art*, pages 128–143. Springer, 2017.

[111] Cheng Zhi Anna Huang, Tim Cooijmans, Adam Roberts, Aaron Courville, and Douglas Eck. Counterpoint by convolution. *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017*, pages 211–218, 2017.

[112] Li Chia Yang, Szu Yu Chou, and Yi Hsuan Yang. Midinet: A convolutional generative adversarial network for symbolic-domain music generation. *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017*, pages 324–331, 2017.

[113] Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical latent vector model for learning long-term structure in music. *35th International Conference on Machine Learning, ICML 2018*, 10:6939–6954, 2018.

[114] Yu-Siang Huang and Yi-Hsuan Yang. Pop music transformer: Generating music with rhythm and harmony. *arXiv preprint arXiv:2002.00212*, 2020.

[115] Jen-Yu Liu, Yu-Hua Chen, Yin-Cheng Yeh, and Yi-Hsuan Yang. Unconditional audio generation with generative adversarial networks and cycle regularization, 2020.

[116] Christopher Harte, Mark Sandler, and Martin Gasser. Detecting harmonic change in musical audio. In *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia*, AMCMM '06, page 21–26, New York, NY, USA, 2006. Association for Computing Machinery.

[117] Heiga Zen, Keiichi Tokuda, and Alan W. Black. Statistical parametric speech synthesis. *Speech Communication*, 51(11):1039 – 1064, 2009.

[118] H. Ze, A. Senior, and M. Schuster. Statistical parametric speech synthesis using deep neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7962–7966, 2013.

[119] X. Wang, S. Takaki, and J. Yamagishi. Neural source-filter-based waveform model for statistical parametric speech synthesis. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5916–5920, 2019.

[120] R. Prenger, R. Valle, and B. Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621, 2019.

[121] Li-Chia Yang and Alexander Lerch. On the evaluation of generative models in music. *Neural Computing and Applications*, 32(9):4773–4784, 2020.

[122] Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. Openface: An open source facial behavior analysis toolkit. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–10, 2016.

[123] Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs. 2017.

[124] Leo Anthony Celi, Roger G Mark, David J Stone, and Robert A Montgomery. "big data" in the intensive care unit. closing the data loop. *American journal of respiratory and critical care medicine*, 187(11):1157, 2013.

[125] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[126] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.

[127] Jose Roberto Ayala Solares, Francesca Elisa Diletta Raimondi, Yajie Zhu, Fatemeh Rahimian, Dexter Canoy, Jenny Tran, Ana Catarina Pinho Gomes, Amir H. Payberah, Mariagrazia Zottoli, Milad Nazarzadeh, Nathalie Conrad, Kazem Rahimi, and Gholamreza Salimi-Khorshidi. Deep learning for electronic health records: A comparative review of multiple deep neural architectures. *Journal of Biomedical Informatics*, 101(November 2019):103337, 2020.

[128] Cao Xiao, Edward Choi, and Jimeng Sun. Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. *Journal of the American Medical Informatics Association*, 25(10):1419–1428, 2018.

[129] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[130] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[131] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[132] Pranjul Yadav, Michael Steinbach, Vipin Kumar, and Gyorgy Simon. Mining electronic health records (ehrs): A survey. *ACM Comput. Surv.*, 50(6), January 2018.

[133] Alexandre Yahi, Rami Vanguri, Noémie Elhadad, and Nicholas P. Tatonetti. Generative Adversarial Networks for Electronic Health Records: A Framework for Exploring and Evaluating Methods for Predicting Drug-Induced Laboratory Test Trajectories. (Nips):1–11, 2017.

[134] Jiaqi Guan, Runzhe Li, Sheng Yu, and Xuegong Zhang. Generation of synthetic electronic medical record text. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 374–380. IEEE, 2018.

[135] Kun Ouyang, Reza Shokri, David S. Rosenblum, and Wenzhuo Yang. A non-parametric generative model for human trajectories. *IJCAI International Joint Conference on Artificial Intelligence*, 2018-July:3812–3817, 2018.

[136] Ziheng Lin, Mogeng Yin, Sidney Feygin, Madeleine Sheehan, Jean-Francois Paiement, and Alexei Pozdnoukhov. Deep Generative Models of Urban Mobility. *ACM SIGKDD Conference*, (17), 2017.

[137] Z. Liu, Z. Li, K. Wu, and M. Li. Urban traffic prediction from mobility data using deep learning. *IEEE Network*, 32(4):40–46, 2018.

[138] T. Donna Chen, Kara M. Kockelman, and Josiah P. Hanna. Operations of a shared, autonomous, electric vehicle fleet: Implications of vehicle and charging infrastructure decisions. *Transportation Research Part A: Policy and Practice*, 94:243 – 254, 2016.

[139] Ezzeddine Fatnassi, Jouhaina Chaouachi, and Walid Klibi. Planning and operating a shared goods and passengers on-demand rapid transit system for sustainable city-logistics. *Transportation Research Part B: Methodological*, 81(P2):440–460, 2015.

[140] Fengli Xu, Zhen Tu, Yong Li, Pengyu Zhang, Xiaoming Fu, and Depeng Jin. Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1241–1250, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee.

[141] V. Bindschaedler and R. Shokri. Synthesizing plausible privacy-preserving location traces. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 546–563, 2016.