Synthetic Data - what, why and how?

James Jordon
jjordon@turing.ac.uk

Florimond Houssiau
fhoussiau@turing.ac.uk

Giovanni Cherubin
gcherubin@turing.ac.uk

Samuel N. Cohen
scohen@turing.ac.uk

Lukasz Szpruch
l.szpruch@ed.ac.uk

Mirko Bottarelli
mirko.bottarelli@warwick.ac.uk

Carsten Maple
cm@warwick.ac.uk

Adrian Weller
aweller@turing.ac.uk

The Alan Turing Institute

THE ROYAL SOCIETY

This report was commissioned by the Royal Society.

Executive Summary

This explainer document aims to provide an overview of the current state of the rapidly expanding work on synthetic data technologies, with a particular focus on privacy. The article is intended for a non-technical audience, though some formal definitions have been given to provide clarity to specialists. This article is intended to enable the reader to quickly become familiar with the notion of synthetic data, as well as understand some of the subtle intricacies that come with it. We do believe that synthetic data is a very useful tool, and our hope is that this report highlights that, while drawing attention to nuances that can easily be overlooked in its deployment.

The following are the key messages that we hope to convey.

Synthetic data is a technology with significant promise. There are many applications of synthetic data: privacy, fairness, and data augmentation, to name a few. Each of these applications has the potential for a tremendous impact but also comes with risks.

Synthetic data can accelerate development. Good quality synthetic data can significantly accelerate data science projects and reduce the cost of the software development lifecycle. When combined with secure research environments and federated learning techniques, it contributes to data democratisation.

Synthetic data is not automatically private. A common misconception with synthetic data is that it is inherently private. This is not the case. Synthetic data has the capacity to leak information about the data it was derived from and is vulnerable to privacy attacks. Significant care is required to produce synthetic data that is useful and comes with privacy guarantees.

Synthetic data is not a replacement for real data. Synthetic data that comes with privacy guarantees is necessarily a distorted version of the real data. Therefore, any modelling or inference performed on synthetic data comes with additional risks. It is our belief that synthetic data should be used as a tool to accelerate the "research pipeline" but, ultimately, any final tools (that will be deployed in the real world) should be evaluated, and if necessary, fine-tuned, on the real data.

Outliers are hard to capture privately. Outliers and low probability events, as are often found in real data, are particularly difficult to capture and include in a synthetic dataset in a private way. For example, it would be very difficult to "hide" a multi-billionaire in synthetic data that contained information about wealth. A synthetic data generator would either not accurately replicate statistics regarding the very wealthy or would reveal potentially private information about these individuals.

Empirically evaluating the privacy of a single dataset can be problematic. Rigorous notions of privacy (e.g differential privacy) are a requirement on the *mechanism that generated* a synthetic dataset, rather than on the dataset itself. It is not possible to rigorously evaluate the privacy of a given synthetic dataset by directly comparing it with real data. Empirical evaluations can prove useful as tools to detect possible flaws in an algorithm or its implementation but may lead to false claims of privacy when there is none.

Black box models can be particularly opaque when it comes to generating synthetic data. Overparametrised generative models excel in producing high-dimensional synthetic data, but the levels of accuracy and privacy of these datasets are hard to estimate and can vary significantly across produced data points.

Synthetic data goes beyond privacy. Synthetic data provides promising tools to improve fairness, bias and the robustness of machine learning systems, but significantly more research is required to fully understand the opportunities and the limitations of this approach.

Contents

1 Introduction

The availability of high volume, high velocity and high variety datasets, together with advanced statistical tools for extracting information, has the potential to improve decision-making and accelerate research and innovation. At the same time, many large-scale datasets are highly sensitive (e.g. in health or finance) and sharing them may violate fundamental rights guarded by modern privacy regulations (e.g. GDPR or CCPA). A large number of real-world examples demonstrate that high-dimensional, often sparse, datasets are inherently vulnerable to privacy attacks and that existing anonymisation techniques do not provide adequate protection. This limits our ability to share these large datasets, creating a bottleneck on the development and deployment of machine learning and data science methods.

Synthetic data is generated by a model, often with the purpose of using it in place of real data. By controlling the data generation process, the end-user can, in principle, adjust the amount of private information released by synthetic data and control its resemblance to real data. As well as addressing privacy concerns, one can to adjust for biases in historical datasets and to produce plausible hypothetical scenarios.

If used responsibly, synthetic data promises to enable learning across datasets when the privacy of the data needs to be preserved; or when data is incomplete, scarce or biased. It can help researchers and developers prototype data-driven models and be used to verify and validate machine learning pipelines, providing some assurance of performance. It can also fuel responsible innovation by creating digital sandbox environments used by startups and researchers in hackathon-style events.

Each of these uses presents great opportunities, but also challenges that require tailor-made solutions. Synthetic data generation is a developing area of research, and systematic frameworks that would enable the deployment of this technology safely and responsibly are still missing.

1.1 Report Structure

This explainer is organised as follows. In Section 2 we introduce a definition for synthetic data, give a brief history of its inception, and begin to answer one of the core questions surrounding synthetic data: *can it replace replace real data?* In Section 3, we introduce key machine learning applications for synthetic data.

Sections 4-7 are dedicated to private synthetic data generation. In Section 4, we introduce privacy from a more general perspective than synthetic data, introducing differential privacy and discussing some of its limitations. In Section 5 we discuss three key attributes for evaluation of private synthetic data: utility, fidelity, and privacy. In Section 6, we discuss empirical evaluations of synthetic data for these key attributes. In Section 7, we discuss key differences between general privacy, and privacy as applied to the generation of synthetic data. We also survey existing methods in the space of private synthetic data generation. At the end of the section we discuss partially synthetic data, in which synthetic data is generated to create a hybrid real-synthetic dataset.

In Sections 8 and 9 we discuss synthetic data for fairness and data augmentation. In Section 10, we provide a more in-depth survey of existing generative models. Finally, in Section 11, we summarise key themes from discussions with industry partners and start-ups in the field of synthetic data.

2 What is Synthetic Data?

Despite tremendous interest in synthetic data, [1–5], to the best of our knowledge there is no widely accepted definition. In order to encapsulate the full breadth of applications and approaches to synthetic data, we propose the following definition.

Definition 1 *Synthetic data is data that has been generated using a purpose-built mathematical model or algorithm, with the aim of solving a (set of) data science task(s).*

We contrast synthetic data with real data, which is generated not by a model but by real world systems (e.g financial transactions, satellite images, medical tests etc.). The *model* – the synthetic data generator – can take many forms, from deep learning architectures such as the popular Generative Adversarial Networks (GANs) [6], or Variational Auto-encoders (VAEs) [7], through agent-based and econometric models [8], to a set of (stochastic) differential equations modeling a physical or economic system [9].

Using computer-generated synthetic data to solve particular tasks is not a new idea, and can be dated back at least as far as the pioneering work of Stanislaw Ulam and John von Neumann in the 1940s on Monte Carlo simulation methods. Synthetically generated data has been widely used in research, as it provides a 'ground truth', which is very useful in developing and evaluating machine learning pipelines.

The recent increase in data protection regulations has fueled the use of synthetic data to mitigate disclosure risk. The key hope, which goes back to work by

Rubin and Little, [10–12], is to be able to use synthetic data in place of real data, to avoid privacy concerns [13–17].

The need to extract actionable information from large datasets led to the development of complex, data-driven (machine learning) models. For these models, the role of data in driving model selection is more prominent than it is for simpler, handcrafted models. This means that the quality of the model's output is directly dependent on the quality of the data used to train these models. This leads to a number of uses for synthetically generated data. One such use is bias removal (e.g. historical biases in gender or race) [18–23]. Given biased training data, a natural approach is to train models using available data; these biases can then be seen in the output of the trained models. Rather than attempting to de-bias each trained model individually, one could generate a de-biased synthetic dataset and use it to train each model [22, 23], creating a unified approach for handling biases across an organisation. Another use would be to use synthetic data to enlarge datasets that are too small, e.g. to provide robustness against "outlier" examples [24–29]. Another key use case, to which we pay particular attention in this report, is the goal of using synthetic data to protect privacy. In each of these cases, the goal is to create synthetic data which resembles some aspects of the real data but not others. To maximise the utility of synthetic data, a fine balance must often be struck between competing objectives.

It is crucial to understand that synthetic data does *not* automatically address any of these problems. Training an off-the-shelf generative model based on real data, and then using this trained model to generate synthetic data, is *not inherently private*. Standard GANs do not generate private nor unbiased data. In fact, machine learning models have demonstrated the capability to (undesirably) memorise their training inputs [30, 31]. Applied to GANs, this can result in memorisation and regurgitation of the training data [32], undermining privacy in the synthetic data. At the other extreme, synthetic data can be generated without training data, e.g. using agent-based models that mimic the data generation process, such as agents transacting in a financial network. With no access to any real data, the synthetic data generator is private, but the data it generates is limited to the model's predetermined configuration, and will not enable statistical inferences to be reliably drawn about the real world.

2.1 Can synthetic data replace real data?

The use of synthetic data raises two key questions:

1. Can we do the same things *with* synthetic data that we do *with* real data?
2. Can we do the same things *to* synthetic data that we do *to* real data?

The sorts of things we may wish to do *with* synthetic data are building models, performing data analysis, testing hypothesis, etc. Things one may wish to do *to* synthetic data, for example, might be linking separate datasets together, or extending a synthetic dataset when new records are added to the original dataset.

2.1.1 How should we approach doing things *with* synthetic data?

Ideally, one may hope that synthetic data can be simply plugged in wherever one might usually use real data (e.g. as training data for a model). Many papers on synthetic data evaluate it in this way. However, with private data, a more careful approach may lead to more accurate information being extracted from the synthetic data [33]. In particular, conclusions from data analysis and hypothesis testing are necessarily weaker when using synthetic rather than real data, and the statistical significance of such analyses needs to be adjusted accordingly.

A particular concern for private data is bias. Ghalebikesabi et al. [34] warn against the risks of learning from synthetic data, and propose a methodology for learning unbiasedly from such data. Wilde et al. [35] demonstrate superior performance when model parameters are updated using Bayesian inference, rather than approaches that fail to account for the fact the training data is synthetic.

2.1.2 Data Linking

Something that can naturally be done with real data is linking. One dataset may contain an individual's lab test results, another may contain their genetic data, and another their hospital appointments. Each of these datasets can be linked to create a larger dataset containing information about inter-dataset correlations. If these datasets were synthesised independently, the 1-1 match between datasets will be broken; if, in the future, someone wished to pull together these synthetic datasets to investigate the correlations between, say, genetic data and lab test results, they would not be able to do so effectively.

One solution would be to encourage data holders to generate synthetic data with other (previously generated) synthetic datasets in mind. This may be appropriate in some situations (e.g. when the two datasets are being held by the same data holder), but in general this will not be the case. Moreover, the initial privacy loss suffered by an individual present in both datasets will be greater than if synthetic data was generated independently. This is particularly inefficient when linking the datasets might not be important, or the benefits of doing so are unclear.

In these situations, there is a need to be able to link two independently generated synthetic datasets (given access to real data) in a minimally privacy-leaking way. One workaround would be to simply generate a new joint dataset from the newly-joined underlying real datasets; but this does not leverage the existing synthetic data. A less naive approach would be to conditionally generate one of the two synthetic datasets based on the other (existing) synthetic dataset. This is a reduction in privacy cost over generating from scratch, but still fails to leverage the second already-generated synthetic dataset.

2.2 Combining Synthetic Data with Other Technologies

Secure Research environment. Synthetic data, in particular with differential privacy, has a natural application within secure research environments, in which decreasingly private data can be accessed in increasingly more “secure” environments. Consider releasing a dataset with strong privacy guarantees initially, evaluating a range of machine learning methods, then selecting the top N candidates, and giving them access to less private data. This can be repeated, giving a ‘tournament’ of methods. This raises the question of how to generate a series of datasets $\mathcal{D}_1, ..., \mathcal{D}_n$ which decrease in privacy individually and when taken together (so any subset $\mathcal{D}_1, ..., \mathcal{D}_j,\ j < n$ is as private as the terminal dataset, $\mathcal{D}_j$).

Federated learning. Federated learning is an emerging technology that enables training across decentralised datasets without pooling these datasets together. This contrasts with traditional centralised machine learning techniques, where the data is uploaded to one server, as well as to classical decentralised approaches which often assume that local data samples are identically distributed. In federated learning, distributed data holders allow an algorithm to be run on their private data, and only the (possibly noisy) outputs are released, without giving direct access to the data. The challenge with it is that, without accessing the data first, it might not be clear what algorithm one should run. Developing an algorithm on private synthetic data samples and evaluating its utility on real (distributed) data seems a very promising approach to this data bottleneck.

3 Why use Synthetic Data?

Synthetic data is being used as a solution to a variety of problems in many domains. Three key areas that are of particular interest in a machine learning context are: (i) private data release (Section 4); (ii) data de-biasing and fairness (Section 8); and (iii) data augmentation for robustness (Section 9). Although these are the areas that appear to have the most promise, this list is not exhaustive. Before going into details for each of them, we outline the key ideas for these areas and some of the specific use cases (and non-use cases) below.

3.1 Private Data Release

The wide adoption of data-driven machine learning solutions as the prevailing approach to innovate has created a need to share data. Without access to quality data, scientists and developers cannot make meaningful progress. However, GDPR, HIPAA, and a host of privacy regulations require data on individuals not to be shared carelessly (rightly so). The result is typically a long series of “jumping through hoops” in an attempt to access the necessary data. Synthetic data offers a potential solution.

Development of ML tools. In this use case, a data controller may wish to assess an ML group's ability to solve a problem, or perhaps even assess several groups simultaneously to select the best partner with which to develop a final solution. In order to avoid privacy concerns, they plan to share synthetic data with their potential partners. For synthetic data to be useful in this setting, model development that is performed on the synthetic data should lead to the same conclusions as if it were carried out on the real data. More concretely, if a researcher comparing two models on the synthetic data were to conclude that model A outperforms model B for a given task, then the same conclusion should be reached when testing both algorithms on the real data. This suggests that, though the synthetic data would need to share many statistical properties with the real data, one can imagine that there are some properties that would not affect these comparisons. Once a final group/model has been determined, it can be taken to the real data for testing, tuning or even a complete re-training.

Software testing. There is a significant appetite for vast amounts of test production data for both system testing and User Acceptance Testing. Synthetic data can remove the requirement of going through lengthy and repeated approvals (e.g GDPR) and sanitation processes and hence save significant time and effort in the development lifecycle. In this setting, it is important that synthetic data used for software testing is semantically correct, but it need not necessarily be statistically correct. Mathematically, this amounts to learning the *support* of the distribution and relevant structural properties (e.g time-series data), but not necessarily the distribution itself. Naturally, by not requiring statistical accuracy, there is much more room for increased privacy (or increased utility at the same privacy level). This is one of the design principles of OpenSafely [36], where practitioners are able to test and develop algorithms on dummy data before running them once on the real data.

Deploying private machine learning tools. Machine learning models are not inherently private. It is well known that neural networks have the capability to memorise training inputs. Membership inference attacks are possible against such networks [37] and, as such, privacy-enforcing training algorithms for machine learning models have been developed [38–41]. An apparent alternative (to enforcing privacy during the training of a model) would be to generate private data and then train a model using this data. *Perhaps* one advantage of such an approach would be that a single private synthetic dataset could create a unified approach to privacy (within a single organisation, say), but we believe that the cost in utility would outweigh the potential "simplicity" of the approach in most applications.

While training the model on private synthetic data might be appealing, it has limitations. Private data generation isn't often able to capture all of the statistical structure that might be important to develop accurate models. As such we believe it to be more prudent to focus efforts on the privacy of trained models, rather than on trying to generate private data with which to train.

3.2 De-biasing

Reducing/removing bias. When generating synthetic data, one can aim at producing samples that do not suffer from historical biases but are otherwise still statistically accurate. Such data can be used then for training 'black box' ML pipelines, while mitigating the risk of historical biases being amplified [42]. Importantly, such data can be reused to train multiple models. This should be contrasted with the approach of correcting each trained model separately. The latter approach has an additional disadvantage, as it could lead to inconsistencies in the way 'fairness and bias' are treated within an organisation. It must be recognised, however, that employing such methods to remove bias from the data introduces additional model risks that need to be quantified and monitored.

What-if-scenario generation. Adjacent to bias removal in datasets is the question of *causal modelling* – i.e. asking the question "What if?". Synthetic data may allow us to explore data generated according to the same causal structure but adjusted distributions, or with different causal interventions placed on the data generating process. One must be very careful to properly model causal relationships though, as causal modelling is sensitive to assumptions and is *not* the same as conditional generation. Indeed, the trustworthy deployment of data-driven models requires that these perform well in situations that differ from the real data. Of course, again, we stress that model risk is being introduced as generative models are being used to produce these new scenarios.

3.3 Data Augmentation

Data labelling. Deep neural networks are state of the art technology in computer vision applications. However, training deep neural networks requires vast amounts of (correctly) labelled data, which is often costly to produce. Synthetically generated labelled data offers a cost-efficient solution to this challenge, and has already been adopted by industry [43]. In this application, one trains a neural network on synthetic data with the intention to deploy it on real data. In general, privacy is not of primary concern in these applications, as the data is not being used to replace the real data but to be used alongside it.

4 Privacy in Machine Learning - An Overview

Privacy is an incredibly large field, with practitioners coming from a wide variety of domains. Here, we present our view on privacy within the context of machine learning. We take a ground-up approach, motivating privacy through the notion of an adversary that has the potential to cause harm should too much information about an individual be revealed to them.

Privacy is a fundamental human right and a prerequisite for freedom of thought and expression. For this reason, a key requirement of privacy is the *consent* of individuals to have their data collected. This consent usually relies on the expectation that the collection of their data, and the subsequent release of information derived from it, will not cause them *harm*. For some types of information, such as an individual's name, address, and phone number, the potential for harm is clear. For others, the potential for harm is more subtle: e.g. it might be that an insurance company would increase the price of an individual's insurance premiums based on knowledge that the specific individual is a smoker, without explicitly requesting this information. This potential for harm is caused by the ability of an *adversary* to gain information about an individual due to the release of data, or more generally from output that is derived from the data (again, this could be a synthetic dataset, or some other algorithm output).

On the other hand, there is a highly social aspect to working with data. Shared datasets that can be used to benchmark models create a community that promotes rapid advancement of technology (see, for example, the rapid progress made in image classification that was caused by the availability of the MNIST and CIFAR datasets [44, 45]). In general, researchers want data, and it may not need to be too accurate for them to start being able to work with it.

Since the late 1990s, much research has been carried out within the realm of privacy. Early on, privacy was often tied together with the notion of *anonymity*, which came in a variety of flavours, from basic name/address/birthday removal ("pseudonymisation"), to k-anonymity [46] (which itself had several iterations [47–49]). Although these notions apply to data, they really only have meaning on the raw data itself, and moreover have been shown to be inadequate even for that [50, 51]. These approaches were built to prevent against *known* attacks. More recently, privacy is being studied with a view to prevent against abstract *threat models* rather than specific instantiations of an attack.

4.1 The Threat Model View

Threat model privacy attacks can be summarised into 3 types: membership inference; attribute inference; and reconstruction attacks.

Threat	**Attacker's knowledge of Targeted Individual**	**Attacker's goal**
Membership inference	Partial/Entire record	Determine if Targeted Individual was in the original data
Attribute inference	Partial record	Recover missing attributes of Targeted Individual's data
Reconstruction attack	N/A	Recover entire records from the original data

These attacks differ both in their goals and in some of the assumptions placed on the adversary's prior knowledge *of the targeted individual*. It should be noted, however, that these attacks do not necessarily place any assumptions on the prior knowledge that an adversary might have about the individuals it is *not* targeting. These attacks are not typically performed against a single individual but multiple individuals at once, with a breach of any individual's privacy being considered a success by the adversary.

Membership Inference. Membership inference [37, 52] aims to determine whether an individual (whose full record might be known to the adversary) was part of the data that was given as input to an algorithm, given the output of the algorithm (and, potentially, knowledge of the workings of the algorithm). On its own, this is only of relevance when membership in the dataset implies some information about an individual. For example, it is not particularly useful (or much of a privacy violation) to ascertain that a particular individual's data was in the 2021 UK Census dataset. Following the example of the smoker, though, it is a violation to be able to ascertain that a given individual is in a dataset that contains only smokers (such as a dataset used in a scientific study of smokers).

Attribute Inference. Attribute inference [53, 54] is slowly becoming infamous as a non-violation of privacy.The goal with attribute inference is to determine some extra information about an individual given some prior knowledge about some of their attributes and access to an algorithm's output (e.g. synthetic data, or a trained ML model). Of course, this is precisely what predictive models aim to do – predict (an) attribute(s) from a set of other attributes [55]. But this leads to an almost paradoxical conclusion, how can a model that was trained *without* an individual's data, violate their privacy? This almost-paradox has led to many privacy researchers abandoning attribute inference as a violation of privacy [56, 57]. However, the question should not be, "does this allow you to learn more about an individual?", but rather, "does this allow you to learn more about an individual than if they had not been in the data?" It is still possible (in fact, with ML it is very likely [58]) that a trained model will perform better on its training inputs than on inputs not used for its training. This indicates that the release of such a model does violate the privacy of the individuals in the training set [59]. Recent work in the space has attempted to determine the feasibility of attribute inference attacks. [60] show that even when membership inference is possible, attribute inference may not be; they do, however, demonstrate that *approximate* attribute inference is possible.

Reconstruction Attacks. Reconstruction attacks [61] aim to extract entire records from the training dataset, based on the output of an algorithm. For instance, Dinur and Nissim showed how a database protected by a question-and-answer system can be reconstructed by an attacker if the level of the noise added to answers is low [61]. Another high profile example is the attack performed by researchers at the US Census bureau on aggregates from the 2010 Census. They were able to retrieve exact records for 46% of the US population using publicly released data [62]. Unlike membership and attribute inference attacks, reconstruction attacks are not *targeted*: they aim to retrieve records for any (or all) records in the original data. This could leverage prior knowledge about *some* of the individuals in the training set (with the remaining unknown individuals being considered the targets). Note that the same caveat as for attribute inference attacks applies: even an algorithm sampling records uniformly at random in $\mathcal{X}$ will reconstruct *some* records in the real data with nonzero probability. Evaluation of a reconstruction attack should therefore be contrastive, i.e. based on the difference of reconstruction likelihood for a record due to their presence in the data (although real-world attacks recover such large fractions of the data that this consideration is often not needed).

4.2 Differential Privacy

Differential Privacy [13, 63], first proposed by Dwork et al. in 2006 [64], is becoming increasingly accepted as a robust, meaningful, and practical definition of privacy [65–67]. Informally, differential privacy requires that an algorithm's (necessarily random) output not differ "too much" between *adjacent* datasets. Intuitively, because the outcome cannot differ significantly, there cannot be too much "information leakage" from the dataset to the algorithm output. The definition rests on some notion of datasets being *adjacent* to each other, and can be used to capture the notion of an individual's data. This adjacency can be defined in different ways depending on the type of data structure. With so-called tabular data, adjacency between two datasets is typically defined to mean that one can be obtained from the other by either the removal/addition (unbounded differential privacy [13]) or replacement (bounded differential privacy [13]) of a row.

Defining this adjacency amounts to deciding precisely what information should be protected [68, 69]. With graph-like data, one could consider either entire nodes (along with all associated edges) to be important, or instead consider only the edges themselves to each be individually important [70, 71]. With tabular data, one might define adjacency as datasets differing in precisely one value, allowing "feature-wise" differential privacy that protects each value contributed to the dataset, rather than rows as a whole. This would allow data contributors to decide to maintain the privacy of some, but potentially not all, of their data.

A particularly important feature of differential privacy is that it is *contrastive* – it compares the outcome of an algorithm when an individual is in the training data to the outcome when the individual is not in the training data, or some similar adjacent perturbation. This idea, that privacy cannot be breached when an individual is not in the data, is crucial in dismissing several more ad-hoc notions of privacy. Crucially, for synthetic data, just because one of the synthetic data points *looks like* one of the original data points does not mean that privacy has been violated – the synthetic point might have been generated even without the original point being present in the training data.

Definition 2 (Differential Privacy [13]) *A* randomized *algorithm, $\mathcal{M}$, is (ε, δ)-differentially private if for all $\mathcal{S} \subset \mathrm{Im}(\mathcal{M})$ and for all neighboring datasets $\mathcal{D}, \mathcal{D}'$:*

$$\mathbb{P}(\mathcal{M}(\mathcal{D}) \in \mathcal{S}) \leq e^{\varepsilon}\mathbb{P}(\mathcal{M}(\mathcal{D}') \in \mathcal{S}) + \delta$$

When $\delta = 0$, $\mathcal{M}$ is said to be pure ε-differentially private.

Intuitively, the key promise of differential privacy is that *any* analysis run on the output of a differentially private procedure will yield approximately the same result whether or not any individual contributes their record to the dataset. This also includes potential harms that could be caused by the publication of potentially sensitive information. For instance, assume that a DP procedure is used to train a ML model to detect a specific disease from sensitive medical records. No operation performed on this model (e.g. inspecting its parameters, applying it to well chosen inputs) can reveal information about individual training records. Hence, it serves as a form of statistical guarantee for individuals that the collection and use of their data will not yield negative consequences (that would not otherwise occur even if the data was not shared[1]). Formally, from a Bayesian point of view, this means that for all potential priors over datasets, the posterior computed after observing the outcome will be similar to the posterior obtained if any one user was removed from the dataset [72].

Bayesian interpretation of Differential Privacy *It is instructive to consider the Bayesian interpretation of privacy guarantees implied by differential privacy, which compares the adversary's prior with the posterior. To that end it is useful to view $\mathcal{D}$ and $\mathcal{D}'$ as a realisation of a random variable $\boldsymbol{\mathcal{D}}$. That way we can model prior knowledge an adversary has about the dataset i.e $\mathbb{P}(\boldsymbol{\mathcal{D}} = \mathcal{D})$. Note that $(\epsilon, 0)$-differential privacy implies a bound on the Bayes factor*

$$\frac{\mathbb{P}(\mathcal{M}(\boldsymbol{\mathcal{D}})|\boldsymbol{\mathcal{D}} = \mathcal{D})}{\mathbb{P}(\mathcal{M}(\boldsymbol{\mathcal{D}})|\boldsymbol{\mathcal{D}} = \mathcal{D}')} \leq e^{\epsilon}\,.$$

This then implies a privacy guarantee on posterior beliefs regarding the value of $\boldsymbol{\mathcal{D}}$ given the output of a differentially private algorithm

$$\frac{\mathbb{P}(\boldsymbol{\mathcal{D}} = \mathcal{D}|\mathcal{M}(\boldsymbol{\mathcal{D}}))}{\mathbb{P}(\boldsymbol{\mathcal{D}} = \mathcal{D}'|\mathcal{M}(\boldsymbol{\mathcal{D}}))} = \frac{\mathbb{P}(\mathcal{M}(\boldsymbol{\mathcal{D}})|\boldsymbol{\mathcal{D}} = \mathcal{D})}{\mathbb{P}(\mathcal{M}(\boldsymbol{\mathcal{D}})|\boldsymbol{\mathcal{D}} = \mathcal{D}')}\frac{\mathbb{P}(\boldsymbol{\mathcal{D}} = \mathcal{D})}{\mathbb{P}(\boldsymbol{\mathcal{D}} = \mathcal{D}')} \leq e^{\epsilon}\frac{\mathbb{P}(\boldsymbol{\mathcal{D}} = \mathcal{D})}{\mathbb{P}(\boldsymbol{\mathcal{D}} = \mathcal{D}')}\,.$$

To put it another way, and due to symmetry between D and D', differential privacy implies that the log-odds cannot change significantly,

$$\left|\log\left(\frac{\mathbb{P}(\boldsymbol{\mathcal{D}} = \mathcal{D}|\mathcal{M}(\boldsymbol{\mathcal{D}}))}{\mathbb{P}(\boldsymbol{\mathcal{D}} = \mathcal{D}'|\mathcal{M}(\boldsymbol{\mathcal{D}}))}\right) - \log\left(\frac{\mathbb{P}(\boldsymbol{\mathcal{D}} = \mathcal{D})}{\mathbb{P}(\boldsymbol{\mathcal{D}} = \mathcal{D}')}\right)\right| \leq \epsilon.$$

[1] A notorious [13] example of this is an insurance company updating premiums based on the result of a study showing correlation between smoking and lung cancer. A client's premium might go up even if their data is not used in the study. Differential privacy here ensures that the result of the study would not change too much whether they give their data or not.

In light of the threat model view introduced in Section 4.1, it should be noted that differential privacy provides provable bounds on the ability of an adversary to perform such attacks [73]. These bounds assume worst-case prior information, knowledge of the algorithms, and computing power of the adversary. Differential Privacy also enjoys several nice properties that have helped with its adoption, such as composability, resistance to post-processing, and plausible deniability [13]. Composability and the post-processing theorem, in particular, allow for differentially private algorithms to be built out of smaller building blocks, which is precisely the driving force behind Differentially Private Stochastic Gradient Descent (DPSGD) [38, 39]. DPSGD has enabled differential privacy to be applied to deep learning architectures. This, in turn, has allowed the development of several DPSGD-driven generative models for synthetic data (see Sec. 7.1).

Flavours of Differential Privacy. Above, we introduced the two most common notions of differential privacy, pure- and approximate-differential privacy (ε and (ε, δ)). There are in fact several relaxations of differential privacy, such as Renyi-Differential Privacy [74]; extensions, such as Label-DP [75]; and even stronger notions, such as local differential privacy [76]. Depending on the task at hand, these notions may be more or less useful than vanilla DP.

4.2.1 Limits of DP

Despite being widely accepted as the best available privacy definition, differential privacy is not without its weaknesses.

Choosing parameters ε, δ. The privacy protection afforded by a differentially private mechanism is controlled by parameters ε and δ. Choosing appropriate values for these parameters is notoriously difficult (in part due to their opaqueness in interpretability), and strongly depends on the context [77–79]. This is further complicated by the fact that many methodologies lack a tight analysis of their privacy, leading to larger-than-necessary noise being injected into the system, hindering utility [80].

Relaxations. Though often celebrated as a strength, the lack of assumptions placed on an adversary's knowledge and capabilities can lead to overly conservative computations, which hinder the utility of the output. Researchers have proposed many relaxations of the original definition (which required $\delta = 0$) [81]. However, the privacy guarantees of these can be hard to understand and model. Similarly to the parameters issue, researchers have started using attacks to compare mechanisms using different definitions of privacy [75].

5 Utility, Fidelity and Privacy of Synthetic Data.

For synthetic data to be *meaningful*, it must be similar to *and* different from the original data in some sense. If synthetic data is being considered, then there is a reason that the original data is inappropriate or inadequate for the task at hand – be it because it is non-private, biased, or too small – and so synthetic data that is too similar to the original data will also suffer from the same problems. The "allowed" similarity (or rather the required non-similarity) will differ from task to task, and constitutes one of the 3 attributes that are fundamental to synthetic data generation: *utility*, *fidelity*, and *privacy*.

Utility: The utility of synthetic data often is determined by its usefulness for a given task or set of tasks. This often involves contrasting the performance of models trained on real vs synthetic data, and might involve inspecting concrete metrics such as accuracy, precision, root mean-squared error, etc.; and/or model fairness properties such as demographic parity, fairness through unawareness, or conditional fairness [23]. Doing so often requires the *Train on Synthetic, Test on Real* (TSTR) paradigm [16] in which models are trained on synthetic data and their performance then evaluated on real data.

Fidelity: Often lumped together with utility, we define fidelity to be measures that directly compare the synthetic dataset with the real one (rather than indirectly through a model, or through performance on a given task). From a high-level perspective, fidelity is how well the synthetic data "statistically" matches the real data. Measures of fidelity are often used because of an underlying intuition that a specific fidelity will correspond to improved performance on a wide range of tasks. In the most general case, full statistical similarity (i.e. matching the distributions of the synthetic and real data), should allow many tasks that would be performed on the real data to be performed on the synthetic. However, such a match is difficult, especially in the presence of privacy requirements [82], and even undesirable in the presence of biases [23]. Rather than seeking a "full" statistical match, one might inspect low-dimensional marginals [83], the syntactical accuracy of the synthetic data [84], or look at the distribution of the remaining features conditional on a feature that is known to be biased in the original data.

Utility vs. Fidelity: Much of the literature on synthetic data, in particular for private synthetic data, focuses on the 2-dimensional trade-off between utility and privacy, folding fidelity into utility. While the two are unavoidably linked, they are not synonymous nor perfectly correlated. In some scenarios, fidelity can be reduced while leaving utility unaltered (or vice versa), potentially "leaving room" for other benefits, for example, improved privacy.

Privacy: The privacy of synthetic data is determined by the amount of information that it reveals about the real data used to produce it. Depending on the use case, different privacy guarantees might be required. For example, internal synthetic data release within a secure environment will typically require less stringent privacy evaluation than data released to the general public. Theoretically sound notions such as differential privacy and its offspring exist, allowing for systematic analysis of the privacy of algorithms used to produce synthetic data. Less is known about the precise meaning of the privacy of a specific synthetic data sample if the data generation method is not revealed, or how to evaluate it, since privacy is typically defined as a statistical property over many instances. Of course, extra care is required to ensure that privacy that has been proven on paper is not lost through sloppy implementation of these algorithms in practice [80].

Privacy vs Fidelity: As a rule of thumb, when fidelity increases, the privacy of synthetic data decreases. This means that, in general, it is impossible to generate private synthetic data that will be useful for all use cases. Instead, one might group potential use cases in terms of the type of fidelity that is required (i.e. which features of the original dataset need to be captured by the synthetic data) and generate multiple synthetic datasets, each with user specified privacy guarantees.

5.1 Synthetic Data Desiderata

A good synthetic data generator (SDG) should simultaneously satisfy the following properties:

1. **Syntactical accuracy**: The generated data should be plausible (e.g. a synthetically generated postcode should exist). However, this also requires that certain structural properties of the data are preserved. For example, with time-series data, one needs to ensure that data points are not generated using information from the future. Similarly, when synthesising financial transnational networks, the underlying graph structure of the data must be preserved.

2. **Privacy**: It should be possible to precisely quantify how much information about the original data is revealed through the releasing of the synthetic sample. How exactly one measures privacy will depend on the specific task at hand. While differential privacy is one popular way of assessing the amount of information release through synthetic data generators, a different notion might be required when the data is sparse or one wants to move away from worst-case bounds.

3. **Statistical accuracy**: It should be possible to precisely quantify the statistical similarity (or lack thereof) between the synthetic and the original data. When measuring statistical accuracy, one might be interested in capturing certain marginal distributions and certain relationships between variables, but not others. A good synthetic data generator should allow for control over this.

4. **Efficiency**: The algorithm should scale well with the dimension of the data space (i.e. feature space). It is well known that, in general, approximation of distributions can suffer from the curse of dimensionality,

and consequently sampling from unstructured distributions is an NP-hard problem.

While it its relatively straightforward to design algorithms for which a subset of these properties hold, there is currently no systematic framework for developing SDGs for which all 4 properties are satisfied simultaneously. For example, generation of statistically accurate but private data is hard, as these goals may be in conflict. Specifically within the realm of differentially private *synthetic data*, Ullman et al. [82] demonstrated that a computationally efficient algorithm (i.e. runs in polynomial time) that generates synthetic data that both: (i) satisfies differential privacy; and (ii) preserves the correlations between pairs of features, does not exist. This result holds in a general sense, in that for every algorithm that *could* generate synthetic data, there is *a* dataset that "will not work". That said, it is possible that:

- for a specific application (e.g., dataset), it is possible to efficiently generate DP synthetic data;
- one may not be interested in the correlations being preserved (i.e. the application may demand a different fidelity notion).

Nevertheless, this impossibility result implies that one needs to assess the privacy and fidelity of the data on a per-case basis. Most importantly, there is no "one-size-fits-all" differentially private synthetic data generation method.

In [85], the authors show that by reducing the requirement that all correlations be matched to the requirement that *most* correlations be matched, a computationally efficient algorithm *does* exist. Such a result is promising for synthetic data, but raises the question of being able to quantify what aspects of the data structure (i.e. which correlations) are *not* being matched.

Such results highlight the need for synthetic data to not try to be too general – synthetic data should be generated with a use case in mind. For a given use case, relevant statistical properties can be preserved, while others can be ignored in the name of creating privacy. Below, we give some concrete examples of such use cases, alongside an application we believe to be a misguided endeavour.

6 Auditing Synthetic Data

In this section, we discuss various approaches for empirically evaluating synthetic data, both in terms of its privacy, and its utility and fidelity.

6.1 Empirically Evaluating the Privacy of Synthetic Data

Given that differential privacy is a theoretical notion of privacy that must be proven, and correctly implemented, to be satisfied, a natural question is to ask whether or not one can verify some notion of privacy for a synthetically generated dataset, or a synthetic data generator, empirically. Given that the goal is to protect against the (abstract) threat models outlined in 4.1, can one "prove" privacy by attempting to perform attacks against a given dataset?

DP verification. As mentioned above, the differential privacy of a synthetic dataset is more precisely a property of the algorithm that generated it, and *cannot* be verified by inspecting the synthetic dataset itself. Researchers have been investigating methods for checking that an algorithm meets DP requirements [86–88]. These methods work either by querying the algorithm in search of violations of the privacy definition, or by running known attacks (e.g. membership inference) against it.

These are useful tools, which can be applied to SDG methods, as a way of testing/understanding their privacy. Perhaps the most useful application is to help us understand what values of ε make the most sense in a specific context. However, using these tools to "prove" differential privacy is not possible, as they are based on statistical analysis of the generating algorithm. What is possible is that one can show, with a certain confidence, that an algorithm is *likely* to be differentially private, but doing so would require sampling many, many times (and more samples would be needed for more complex outputs/algorithms, e.g. when the output is a dataset) from the algorithm with many, many different input datasets. Doing so would be *highly* computationally intractable if any sort of meaningful level of confidence was to be achieved.

Leakage estimation. An alternative option for evaluating the privacy of algorithms is to use *leakage estimation* techniques [89, 90], which stem from the quantitative information flow community [91]. These techniques enable quantifying the privacy of an algorithm with respect to a specific threat model (or adversary). For example, in the context of SDG, this means one could use leakage estimation for assessing the resilience of a method against membership inference or attribute inference attacks, which we described above.

A strong advantage of this approach, is that it does not require any formal analysis of the SDG method; additionally, it can be used for selecting the privacy parameters of a DP algorithm. Another advantage is that some of these methods enable a fully black-box analysis; that is, there is no need to describe the algorithm's internals analytically. One disadvantage is that the leakage estimation analysis is done with a specific threat model (or attack) in mind, although there are ways of capturing many attacks with the same analysis [92]. A second disadvantage is that, in the case of black-box leakage estimation methods, the formal guarantees derived via these approaches make the assumption that we can sample an arbitrary amount of data from the algorithm. Nevertheless, they have been shown to be effective when tackling real-world tasks [93].

Empirical privacy evaluation of datasets themselves. The empirical evaluation of privacy of synthetic data is a nascent and challenging area of research. Despite the fact that *differential* privacy cannot be established for a dataset in isolation, practitioners in the field of synthetic data have made use of a hold-out test set to evaluate (other notions of) the privacy of generated synthetic data [94]. The Nearest-Neighbour distance ratio (NNDR) has been used to inspect whether or not synthetic data points are closer than some hold-out test points to the underlying real data points (on average). This involves splitting the data into a training set, and a test set (as is similarly done in supervised learning problems). The training set is then used to train the model. Once trained, samples are drawn from the trained model, and the distance from these samples to the training data is compared with the distance from the test set to the training data.

Note that although this method is agnostic of the method used to train the data generator, it does rely on a hold-out test set being available. As noted in [95], non-existence of points can be just as revealing as the existence of points in the synthetic dataset, but privacy analysis via NNDR does not capture this behaviour. Indeed it is possible to satisfy NNDR by creating "holes" in the synthetic data around the real data points, but such an approach would reveal where the real records should be.

A similar application of NNDR is performed in [96] to attempt to ensure privacy of the generative model, which also does not control for the creation of such "holes". In [83], they use a nearest-neighbour based classifier to quantify the risk of attribute disclosure but do so in a non-contrastive way, thus rendering the privacy analysis weak. An NNDR-type metric was also used in [97] to assess the relative privacy of several generative models proposed for health data.

Attacks against private synthetic data. As we explain in section 4.1, one of the approaches to understand and analyse privacy is through the lens of attacks: what can a motivated attacker learn about users in the dataset? In addition to being intuitive, this approach can help us evaluate whether a system protects user privacy in a given context, and compare methods built with different privacy definitions in mind. This is particularly relevant for synthetic data generation: as we have presented in previous sections, a wide variety of SDG methods have been proposed, with widely different privacy definitions, choice of parameters, and assumptions. Despite the potential of adversarial approaches to evaluate privacy risks of synthetic data, the development of privacy attacks against SDG remains underexplored. We here review existing attacks, and suggest promising research areas.

The main method to evaluate privacy risks in synthetic data was proposed by Stadler et al. [80] in a recent paper. They propose a general methodology to apply membership and attribute inference attacks on *any* synthetic data generation model. They assume black-box access to the SDG method, and specifically, being able to retrain the SDG model on new data. Indeed, analysing the synthetic data alone (as in NNDR metrics) is, in general, not sufficient to properly understand information leakages: an algorithm sampling records uniformly at random might, by coincidence, replicate exactly some records from some private dataset, but this would not usually be considered a privacy violation. Further, this assumption is a key transparency requirement: in order to audit synthetic data, it is necessary to be able to understand how it was generated. Therefore, privacy guarantees cannot usually be obtained by maintaining secrecy of the generating algorithm. The method proposed by Stadler et al. uses *shadow modelling*: the attacker simulates many runs of the SDG algorithm, using auxiliary data, to generate synthetic datasets trained with or without a target user. A binary classifier is then trained on features extracted from these synthetic datasets to predict whether the target user is in the training dataset. Empirical results suggest that current SDG methods either are vulnerable to this attack or, if the attack fails, lead to an accuracy worse than non-SDG methods for a range of data analysis tasks.

Outside of this specific paper, there is a rich literature on privacy attacks that can be leveraged to develop attacks against synthetic data. We here detail two possible lines of research for this approach:

1. Many synthetic data generation methods rely on Generative Adversarial Networks (see, e.g., [15, 17, 23, 98, 99]). Researchers have demonstrated that such GANs can be vulnerable to white-box and black-box membership inference attacks [100, 101]. A key question is then: how can these attacks be ported to the setup where the attacker has access to *data* generated by the model, rather than the model itself.

2. Some methods built with specific use cases in mind can aim to closely replicate statistical properties of the training dataset, such as one-way marginals histograms or correlations. Many membership inference attacks have been proposed against aggregate statistics, from simple statistical tests [102, 103] to advanced attacks based on shadow models [104]. If the synthetic data accurately reproduces many aggregates from the original data, one can apply these attacks to the synthetic data to infer membership of specific records in the training dataset. This leads to an interesting question: how many statistics can be accurately reproduced from the original data, without enabling such attacks?

These are only two prospective research directions for the adversarial evaluation of synthetic data, which is an open line of research.

6.2 Evaluating the Utility and Fidelity of Synthetic Datasets

The methods presented in this section generally focus on privacy as their primary design goal, most often through explicit guarantees such as differential privacy. In this section, we review approaches to evaluate the secondary goal of such datasets: their *utility* and *fidelity*. The utility of a private synthetic dataset is determined entirely by its application. Generating synthetic data to enable release of otherwise private data has almost as many use cases as there are machine learning problems – any data-driven problem might be derived from sensitive data and the data controllers may wish to investigate which ML methods might address the problem. Below (6.2.1) we give example use-cases, and suggest how utility might be measured in such cases. Fidelity is less well-defined: it generally aims at evaluating how close the *distribution* of the synthetic dataset is to that of the real data, the idea being that if the distributions match, the synthetic data can be used to perform any task as accurately as with the real data. We discuss this more general use case, and review works studying fidelity, in section 6.2.2.

6.2.1 Utility-driven evaluation

The key use-case for privately-generated synthetic data is to enable research and industry data analysis tasks without access to sensitive data. A particularly important application is the development of machine learning (ML) models to perform inference and classification tasks from data. In this setting, the goal is to determine the best model (or a selection of best contenders), and train it (choose its parameters) to perform a given task. In general, such a task will come with its own metric of performance (e.g. accuracy/AUROC in a classification task). There are broadly two directions of research aiming to evaluate the suitability of synthetic data for ML training: (1) evaluating the performance of models trained on synthetic data, and (2) evaluating whether the *relative performances* of different models are similar on synthetic and real data.

The first approach assumes that analysts will train a machine learning model on the synthetic data, and use this model directly on real, future data. In this situation, it is important that the accuracy of a model estimated with synthetic data reflects its accuracy on real data. For instance, Beaulieu et al. evaluate their synthetic data generation method by measuring the accuracy of classifiers trained on synthetic datasets on the real sensitive medical data used to generate the synthetic data [105]. Patki et al. pushed this further, by distributing synthetic datasets and real datasets randomly to teams of data scientists, and evaluating whether teams working on real and synthetic datasets would arrive at approximately the same conclusions [106]. Similar approaches were used by Tao et al. [107], where a XGBoost classifier is trained on synthetic data and evaluated on real data for a range of different tabular datasets.

Note that this approach makes some assumptions on the family of models that will be trained, since it is impossible to test *all possible classes of models*, as well as all possible choices of hyper-parameters. When synthetic data is generated with a set of specific tasks in mind, custom metrics can also be developed that capture the accuracy on these specific tasks. For instance, in the NIST challenge[2], accuracy was measured by the error on the Gini coefficient of incomes and the gender pay gap in (real) demographic data with financial information, when estimated on synthetic data [108].

The second approach studies whether, for a battery of models, their ranking in

[2] `https://www.nist.gov/ctl/pscr/open-innovation-prize-challenges/current-and-upcoming-prize-challenges/2020-differential`

terms of accuracy would be the same when trained on synthetic or real data. This setup assumes that analysts use synthetic data to *select* a model, which is then trained on a real dataset (for better real-world performances). Crucially, the goal is that model development on synthetic data reflects model development on real data – when a comparison is made (e.g. between two choices of hyperparameters), it should mirror the comparison on the real data. The utility of the synthetic dataset is then given by how well the performance ranking of models on the synthetic data matches the ranking that would be determined by the real data. This is challenging to measure since, in theory, one would want to ensure that "all possible methods" are appropriately ranked, including those which have yet to be developed/discovered. One approach might be to approximate this by comparing a list of representative models [109]. It might also make sense to expand the list of representative models by incorporating small variations of each model (i.e. by varying the hyperparameters involved). Efficiently computing this for a broad enough class of models would be key, which may require new insights to ensure the class is indeed sufficiently broad.

An important thing to keep in mind when utilising synthetic data in this way is the variability of various models' performances, especially the variability with respect to the real vs. synthetic data. In particular, if several synthetic datasets (each with decreasing privacy) are going to be used to narrow-down the best methodology, then the process needs to be aware of how the earlier (more private) synthetic datasets will typically create noisier rankings and so the notion of a method being statistically significantly better than another needs to be adjusted accordingly.

6.2.2 Fidelity-driven evaluation

A promise of synthetic data is that it "looks like" real data, and can thus be used for a variety of purposes. From a statistical perspective, the goal that the distribution $|\hat{P}$ used to generate synthetic data is *close to* the (unknown) real data distribution $\mathbb{P}$. Typically, evaluating fidelity in this way involves choosing a distance with which to compare distributions, then evaluating this distance empirically from samples of the real and synthetic datasets.

A simple example is to focus on 1- and 2-way marginals of the data, which can be efficiently computed. The difference between these marginals can be estimated with a wide range of metrics: total variational distance [107], correlations and Cramer's V [107], or classical distances [108]. These metrics aim to capture whether the synthetic data captures basic properties of the real data, such as histograms of individual attributes and relations between pairs of attributes.

Estimating distributional distances in higher dimensions, capturing relations between several attributes at a time, is challenging. Researchers have proposed ad hoc metrics, such as comparing the density of the synthetic and empirical distributions over random subsets of $\mathcal{X}$ [108]. Another solution is the *propensity score* [110, 111], which captures the accuracy of a classifier trained to differentiate real from synthetic data points. Intuitively, if the classifier cannot distinguish the two, then the distribution of synthetic data points must be close to that of real data points (this is the basis of the original GAN framework [6]).

In the context of generative networks (and specifically, GANs), researchers have proposed measures to evaluate the fidelity of synthetic samples. Sajjadi et al. proposed metrics of *precision* (the quality of synthetic samples) and *recall* (the diversity of synthetic samples), inspired by common failure modes of GANs [112]. The key question that these metrics seek to answer is how do the empirical and synthetic distributions *overlap*: precision (resp. recall) captures how much of the synthetic (resp. real) data falls within the support of real (resp. synthetic) data. Researchers have proposed extensions of these metrics to increase their robustness to outliers and make them easier to compute [113] and account for the probability distribution rather than just the support [84].

7 Private Synthetic Data Generation

Thus far, we have focused on privacy in a broader context than synthetic data. This is natural, because most privacy notions apply more generally. However, synthetic data has an additional property that most other (non-synthetic data) outputs do not have – it resides in the space of the real data. That is, the synthetic data takes values from precisely the same space as the original data. This allows us to use synthetic data in the place of real data, and also to compare the similarity of the synthetic and real data directly. We begin by highlighting some key considerations for private synthetic data generation.

The space of datasets can be very high-dimensional. Differential privacy has been conventionally applied to settings in which the dimensionality of the output space is relatively small, such as count queries on rows, classification tasks, etc. Synthetic data generation, on the other hand, gives output in a very high-dimensional space, i.e. the space of datasets (perhaps of a fixed size, N). Releasing such a high-dimensional object is challenging under differential privacy because it leads to higher (worst-case) sensitivity of the generating function (i.e. the algorithm mapping the input dataset to the output dataset).

Because of this increased worst case sensitivity, accurately constructing a dataset under differential privacy (according to some notion of accuracy) is likely to require a large privacy budget. A dataset of 1 million records, each with only 20 features, results in a 20 million dimensional output. Not only is the sensitivity of such a function likely to be high, but even trying to analyse the sensitivity can be an incredibly difficult task. By instead aiming to create a *private generator*, one can alleviate some of the difficulties - the complexity of the generator should not need to scale with the number of rows, but instead only with the dimensionality of the data (e.g. the number of columns in tabular data).

Privacy is a property of the Algorithm, *not* of the Data. An important-to-note property of differential privacy that it is a notion that is concern with probabilistic properties of the generated outputs and not a single realisation/output of the generator. A single output of the generator is is neither private, nor non-private. In a non-synthetic data setting this may be more obvious. If you query the average age of individuals in a dataset, and are given the response '35', without being told how this was computed, you do not know (even with access to the original data) whether the number 35 was computed privately or not. The algorithm that computed this answer could simply be "always answer 35, regardless of input data" which clearly reveals no information. The point is there is nothing private or "unprivate" about the *number* 35.

What *is* private (or non-private) is the algorithm that produces the number, or in the case of this report, that produced the synthetic dataset. Crucially, this means that, at least from the perspective of differential privacy, it is *meaningless* to talk about the privacy as a property of a concrete synthetic dataset.

Private Data vs. Private Generator. The most common approach to generating private synthetic data is to first train a *private generator* (e.g., [14, 15, 17, 114, 115]) on the real data. This model is then directly sampled from to generate individual datapoints and thus build up an entire synthetic dataset. Due to the post-processing theorem [13], if the procedure used to train the private generator is ε-differentially private (ε-DP), it can be used to generate arbitrarily many synthetic data records without affecting the privacy guarantees. In fact, the procedure to generate the synthetic dataset (of arbitrary size) is itself ε-DP, which guarantees that individual records in the real data are protected from attacks.

However, training an $\varepsilon-$DP generator to then generate some fixed number of synthetic samples can lead to overly conservative privacy guarantees, and thus lower utility. Intuitively, DP restricts the amount of information extracted from the real dataset when computing the output (the generator or dataset). The generator acts as a bottleneck for information: any finite sample from a generator necessarily has less information from the original dataset, and thus typically results in lower utility.

Outliers and Fairness. Capturing outliers with private synthetic data is difficult. Outliers are precisely data points with *some* uniquely identifying features; this means that "hiding them in the masses" becomes impossible. In some scenarios, for example credit card fraud detection, detecting outliers is the goal. In such a setting, private synthetic data is unlikely to provide much utility, as the outliers will necessarily be suppressed by the need for privacy. Indeed, [116] posit that if outliers are to be captured, then an SDG method cannot attain both a high privacy and a high utility.

This has a problematic implication for fairness: minority groups, like outliers, can often end up being under-represented in synthetic data [117, 118]. Indeed, there is a clear tension between fairness and privacy, with fairness requiring that there is a good utility, even for minority groups, and privacy hiding the contributions of individuals, and thus collectively hiding the contribution of a minority group. Indeed, [119] posit that current mechanisms are unable to simultaneously achieve privacy, fairness and utility.

7.1 Existing Methods and Technologies

Much attention in the machine learning community is being given to the problem of *generative modelling*. The goal of generative modelling is to generate samples with similar statistical properties to the available training data. Generative models are a key ingredient in synthetic data generation, but crucially require additional thought beyond their basic capability to "generate samples" - the problem of generating samplings from a distribution given training data is underspecified, and can be satisfied by memorising the training data and regurgitating when asked. We defer discussion of generative modelling in general to section 10, but draw attention to existing *privacy-preserving* generative models here.

As already noted in section 4.2, a key ingredient driving many of the deep learning based algorithms is DPSGD [38] (differentially private stochastic gradient descent), which enables differentially private training of general neural network based architectures. Though not designed specifically for generative models, DPSGD can be applied to both GANs [15, 16, 114, 115], and VAEs [120]. Other approaches leverage the popular PATE mechanism [40], which can be applied to black-box models to create a private predictor. This has been used to replace the discriminator in a GAN model with a private PATE model [17], and as a means of passing gradients from discriminator to generator [98]. [121] use a subsample-and-aggregate approach (similar to PATE) alongside differentially private expectation-maximisation (DP-EM) [122].

Other popular approaches involve representing the data in a simple, low-dimensional form, such as using a Bayesian network to represent the data generation process as a series of low-dimensional marginal distributions [14], or leveraging the classical copula framework [123] to learn the generation process [124]. The recent NIST competition[3], was won with an algorithm that learns all 2-way distributions (marginals) in a differentially private way, and then does post-processing to generate data from these (potentially inconsistent) marginals [5]. See also [125] for similar ideas.

As should be clear from the proceeding discussion, there are many ways to enforce privacy, even when using the same underlying algorithm (e.g. a GAN). The question of where and how to enforce privacy is very much open. Intuitively, one wants to apply privacy simultaneously: (i) as close as possible to the output (e.g. to the generator of a GAN rather than the discriminator); (ii) wherever the tightest analysis of the privacy can be done (with GANs, the discriminator

[3] https://www.nist.gov/ctl/pscr/open-innovation-prize-challenges/current-and-upcoming-prize-challenges/2020-differential

is typically easier to analyse). With GANs, we see examples of privacy being applied to the discriminator [15, 17, 114], and to the generator [98].

7.2 Partially Synthetic Data

For tabular data, where a user's data is a n-tuple of *attributes* an alternative to the common flavour of SDG is *partially synthetic data.* In partially synthetic data, the attributes of user records are divided into two categories: *quasi-identifiers* and *sensitive attributes* (formally $\mathcal{X} = \mathcal{X}_{\text{quasi}} \times \mathcal{X}_{\text{sensitive}}$). The quasi-identifiers are assumed to be non-sensitive and can be disclosed unchanged, while the sensitive attributes need to be protected. Partially synthetic data is obtained by first fitting a statistical model $f : \mathcal{X}_{\text{quasi}} \rightarrow \mathcal{X}_{\text{sensitive}}$ on the quasi-identifiers to predict the value of the sensitive attribute, then replacing the sensitive attributes of the data by values produced by the statistical model, a process known as multiple imputation [126]. The synthetic data generated thus contains the quasi-identifiers of all real records. Researchers have proposed partially SDG models using decision trees, support vector machines, and random forests [127, 128] (see the synthpop package [129]).

Utility-wise, partially synthetic data have a clear advantage over *fully* synthetic data, as they not require to estimate the full joint distribution $\mathbb{P}(\mathcal{X})$ but only the conditional distribution $\mathbb{P}(\mathcal{X}_{\text{sensitive}}|\mathcal{X}_{\text{quasi}})$. However, some reports suggest that the utility of such data in practice is not particularly appealing, and that they are probably best suited for preliminary data analyses [130], or when combined with restricted access to confidential data [131].

Privacy-wise, there is the issue that revealing quasi-identifiers might be considered sensitive in some contexts (and this approach is trivially vulnerable to membership inference attacks). Researchers have shown that current methods are vulnerable to record linkage attacks [132, 133], where an attacker identifies which record belongs to a target user using the sensitive attributes. Partially synthetic data generation methods are generally heuristic methods, and do not satisfy guarantees of privacy. Label DP [134] is an extension of differential privacy that fits this context well.

8 De-biased Synthetic Data Generation

Machine learning models are known to inherit biases from their training data [135–138]. De-biasing trained models requires expert knowledge of the model [139, 140], and also an understanding of the different notions of fairness that one may wish to achieve (e.g. fairness through unawareness, demographic parity, conditional fairness [23]). An alternative approach that is being explored is to learn to de-bias the dataset itself, thus creating so-called *fair data*[18–21].

This data de-biasing can be viewed as a sort of synthetic data generation, in which the synthetic data is the de-biased data. Some approaches take it a step further and explicitly model the problem of data de-biasing as a one of "ground-up" generation [22, 23]. These approaches aim to learn a generative model that itself is fair. In [23], they explore several notions of fairness and, via causal modelling, identify strategies for generating data that satisfy the given notions.

8.1 Notions of Fairness

As we have seen, identifying a meaningful, interpretable notion of privacy is difficult, and the same is true for fairness. These are both complex ethical questions that the machine learning community must address sooner rather than later. Unlike privacy, however, obvious notions of fairness *do exist*, and are enforceable in a meaningful way[4]. Each notion typically requires some notion of a set of *protected attributes*, along which fairness must be ensured (e.g. gender, race).

Fairness Through Unawareness (FTU) [23, 141] requires that the protected attributes, and only the protected attributes, not be used by the predictor. This aligns with the idea that two equally qualified people deserve the same job opportunities, independent of race or gender [23]. FTU, however, fails to take into account the effect that protected attributes might have on other unprotected attributes, such as an individual's race resulting in them not being afforded the same educational opportunities as an individual of a different race (and thus resulting in them appearing to be disparately qualified).

Definition 3 (Fairness Through Unawareness [23]) *A predictor $f : \mathcal{X} \to \mathcal{Y}$ is fair if and only if protected attributes $\mathcal{A} \subset \mathcal{X}$ are not explicitly used by f to predict $Y \in \mathcal{Y}$.*

Demographic Parity (DPa) [142], instead, requires that a predictors output's not be correlated with the protected attributes. Indeed, with FTU, an attribute that is correlated with the protected attributes can be used as input to a predictor and thus the predictor can indirectly be correlated with the protected attributes (despite not having direct access to them). DPa is a significantly stronger notion of fairness than FTU, which requires adjusting the distributions of *all* variables that are correlated with the protected attributes.

Definition 4 (Demographic Parity [23]) *A predictor $f : \mathcal{X} \to \mathcal{Y}$ is fair if and only if protected attributes $\mathcal{A} \subset \mathcal{X}$ are independent of the predictions. That is, given a random variable $X \in \mathcal{X}$, let $A \in \mathcal{A}$ be the components of X that are protected. Then f satisfies demographic parity if and only if $f(X)$ is independent of A.*

Under the graphical model approach used in [23], for example, DPa is ensured by deleting all edges that originate from a variable that has a protected attribute anywhere in its causal predecessors. Naturally, such an approach can

[4] This contrasts with the fact that, with privacy, the notion of not wanting your data to affect the output *at all* is not achievable without simply ignoring your data completely.

significantly degrade performance, as many of these variables can be useful predictors of the target. A trade-off is present, in which some fairness may need to be sacrificed for performance, and vice-versa.

8.2 Limitations of Fair Synthetic Data Generation

Though one might hope that fair synthetic data would lead directly to fair predictors (which *is* the case with private synthetic data and private predictors), this is not the case. Of particular importance to note is the fact that a predictor's fairness is *with respect to* a distribution of the features. Indeed, a predictor that is trained on synthetic data may not longer be fair when moved to real data due to a shift in the distribution of the real features. This is partly the reason that [23] take such an extreme approach in removing all contaminated features (rather than trying to only remove the influence of the protected attributes on contaminated unprotected ones).

Moreover, a synthetic dataset's fairness is defined through a given predictor. Giving a more general definition for fair synthetic data, and determining whether or not predictors trained on such a synthetic dataset would be fair, are open problems. The hope that an organisation might make a single "fair" synthetic dataset for use across an organisation would require advancements in this space, requiring a shift in thinking from fair predictors to fair data.

8.3 Existing Methods

As noted above, fair synthetic data generation is a young field. While there is significant amounts of work on creating fair predictors (see e.g. [143] for a recent survey), the work on synthetic data for fairness is more limited.

[23] take a causal, GAN-based approach, using several GAN networks, along with an assumed known causal graph to learn the generative process of the data. Armed with the causal graph, they then generate synthetic data by selectively dropping edges from the model depending on the notion of fairness being targeted.

[22] use a more indirect approach to ensuring fairness. Again based on a GAN model, they instead opt to introduce an additional loss term that penalises disparity between protected attributes taking different values. This means that the learned model might still generate unfair data, but only if doing so results in an increase in the fidelity of the generated synthetic data. This trade-off is controlled by a hyperparameter than can be increased to more stringently ensure fairness.

8.4 Evaluating Utility and Fidelity

Much of the discussion found in Section 6.2 can be applied to fair synthetic data. An additional consideration will necessarily be whether or not the bias has been successfully removed, metrics for which can be found in [23]. These typically involve a comparison between outputs of a predictor trained on the synthetic data when the input protected attributes are varied.

9 Data Augmentation

Perhaps the most successful use case (so far) for synthetic data has been for data augmentation - using synthetic data to enlarge datasets with additional samples to use for training [24, 28, 29]. This is often referred to as semi-supervised learning. The intuition is that synthetic data can act as a regulariser, reducing variance in the learned downstream model [29]. Fortunately, there are several good surveys for data augmentation, and so we defer the reader to those for a more thorough background: time-series data [144]; image data [145].

Of note is the fact that synthetic data driven techniques are more important in domains with less structure (such as with generic tabluar data). In the image domain, there is significant structure that can be exploited to create additional data, such as small rotations, image-flipping, cropping, etc. This structure often does not have a parallel in generic tabular data. As such, augmentation methods driven by synthetic data generators are a promising approach to fill this gap.

9.1 The Basic Principle

The key idea driving the use of synthetic data for data augmentation is that of *generalisability.* The goal of a model is not just to perform well on its training data, but also on data that has not been seen before by the model [146]. The hope with synthetic data is that one can take the available training data and learn a generative model, such as a GAN [25]. This generative model will then be able to produce “realistic” samples of training data, and, *hopefully* these samples will be sufficiently different from the original training data so as to be useful additional data points for the training of the model.

It is particularly important to stress that these samples need to be *sufficiently different* from the original data. If these samples are too similar, then they will provide no benefit over using the original data on its own for training. As is hopefully clear by now, this need for dissimilarity is a common theme with synthetic data. In the context of data augmentation, this need is less strict; with privacy and fairness, the dissimilarity is a question of ethics and a failure to satisfy it can have both moral and legal repercussions. With data augmentation, this is unlikely to be the case.

9.2 What methods are used?

Depending on the domain in question, a variety of models exist that can perform the task of data augmentation. For the most part, the only real requirement is a generative model that is tailored to the domain in question. In Section 10, we discuss different types of generative models in more detail. Due to the need for (some level of) dissimilarity, training a generative model and hoping for the best may not be sufficient, however. Instead, one may wish to impose restrictions that enable this dissimilarity. The authors in [26] show that learning a perfect generator in a GAN leads to poor semi-supervised learning, but that a bad generator performs well, for example. This idea of dissimilarity is often referred to as a need for *diversity* in the generated samples. Metrics for measuring this exist, such as the score presented in [84].

10 Generative Modelling - An Overview

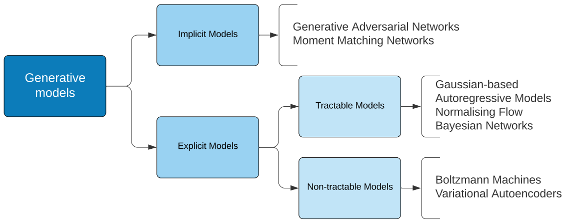

Figure 1: Generative modelling taxonomy based on the maximum likelihood from [147]

Though there is a wealth of *directed* work on synthetic data generation with a specific goal in mind (such as one of the ones discussed in sections 4, 8, and 9), there is also a lot of work focused on generative modelling [for the sake of generative modelling]. The original GAN paper was introduced as a means to generate fake images, not because they wanted to create private, debiased, or even augmented image datasets, but because the task of generating realistic images was hard and the success was easy to evaluate even for non-experts in the field. To that end, there are several methodologies that exist that have yet to be applied to a specific problem.

The remainder of this section is more technical than the rest of this manuscript. Those that wish may skip to Section 11 without any loss of flow.

These methodologies could be taxonomised as in Figure 1, as proposed by Goodfellow [147], focusing on generative models whose parameters are trained to maximise the likelihood of the original data. Such methods can be grouped into two main families, where the underlying density function is either explicitly or implicitly defined. Within explicit models reside statistical methods where new samples are extracted from the distribution arising from the model's definition, which, in turn, must strike a balance between the ability of the model to capture data complexity and to maintain computational tractability [148–151].

Non-tractable density functions can be explicitly tackled through deterministic and stochastic approximations. Variational Auto-Encoders (VAEs) [7] define the loss functions as tractable lower bounds of the non-tractable log-likelihood densities. However, if these deterministic approximations are not carefully calibrated, the model may not converge to the target distribution and consequently generate inconsistent data. On the other hand, stochastic approximations are the basis of Markov chain approaches, where samples are repeatedly drawn after the application of a chosen transition operator. Deep Boltzmann Machines [152] are the main representatives of this class, having all neural units composed of random variables, which simultaneously act as inputs and outputs of the closest layers. Such versatility results in difficulties with training, and thus it is preferable to consider the networks as composed of Restricted Boltzmann Machines (RBMs) [153], consisting of only one visible and one latent layer. In a two-pass learning process, RBMs are progressively trained and then globally fine-tuned. Gibbs-sampling is then used to extract synthetic values.

A completely different direction is taken by implicit generative models, which can be thought of as “black-boxes”, where distributions are not explicitly defined but indirectly revealed through sampling. A first example is Generative Stochastic Networks (GSNs), based on Markov chains. In these networks, the distribution is estimated indirectly, employing a parametric transition operator instead of a parametric model. Nonetheless, they are subject to scalability issues, being not efficiently applicable to high-dimensional scenarios.

Generative Adversarial Networks (GANs) [6] were designed to be jointly parallel and multi-modal (i.e. capable of simultaneously generating multiple valid outputs for the same input). GANs consist of two networks: the generator and the discriminator. These two networks are trained *adversarially.* The generator creates artificial outputs that are passed to the discriminator along with real data. The discriminator is then tasked with identifying which outputs were real, and which were ‘fake’. The final goal here is to reach equilibrium, in which the generated samples follow the same distribution as the real data. When this happens, the discriminator can do no better than random guessing. Theoretically, GAN generators can perfectly imitate the original distribution provided that the network is sufficiently complex enough and the discriminator is optimal [6]. In practive, however, training a standard GAN discriminator to optimality can cause convergence issues and zero-gradients for the generator. Attempts to increase model stability include feature matching, label smoothing, and mini-batch discrimination [154]. Alternatives to the Jensen-Shannon divergence (JSD) (that is implicitly used by standard GANs) have been investigated, such as the Wasserstein distance in WGAN [155], and the Maximum Mean Discrepancy in MMD-GAN [156, 157]. A generalisation of the JSD to f-divergences has also been explored [158]. Unfortunately, no method has proven to be a clear winner [159]. A further risk of GANs lies in mode-collapse, when the generator memorises only a subset of the training information, hence failing to capture the high-level characteristics of the distribution. To tackle this issue, new architectures have been proposed, such as Conditional GANs (CGANs) [160], where additional classification information is provided to both generator and discriminator networks as a form of semi-supervised learning, Deep Convolutional GANs (DCGANs) [161], where convolutional layers substitute pooling layers and Information maximising GANs (InfoGANs) [162] that takes an information-theoretic approach to controlling the generation process. Generative Moment Matching Networks (MMNs) [163] constitute another emerging cluster of implicit models, replacing GANs' discriminators with two-sample tests based on kernel maximum mean discrepancy to measure the distance between modelled and target distributions. Although MMNs offer theoretical guarantees, they are currently outperformed by GANs.

Research on generative models is exploding, both in the evolution of existing models as well as in the introduction of new ones. GAN models are the most popular approach, but their implicit nature necessitates the development of trust through the definition and application of rigorous methodologies and metrics, so far demonstrated to be a difficult task.

10.1 Existing Methodologies

The discussion above focused on generative models in the most general sense, without any consideration for the *type* of data the model needs to generate. Much work in generative modelling has focused on the image domain with GANs, for example, being originally proposed as an image generation framework. Since their inception, however, many works have looked to generalise GANs to other domains. Below we overview some of the leading generative models that exist for a variety of different data types. Figure 2 gives an overview of the taxonomy.

10.1.1 Tabular data

Tabular data consists of values stored in rows and columns, whose synthesis requires the simultaneous modelling of distinct column distributions, as well as row-wise and table-wise constraints. Receiving less attention than image data in the generative domain, tabular data generation still has many obstacles to overcome. Initial generators relied on classifiers, as in the case of inverted decision-trees, vector machines and random forests, which struggle to strike a balance between classifier's accuracy and the risk of leaking information [164–166]. On the other hand, the application of GANs required the conversion of categorical, discrete columns to a continuous form using auto-encoders or through the decomposition in a variable number of Gaussian modes [167–169]. Nonetheless, the independent generation of column values might result in invalid rows, whose semantic correctness requires either the training of additional classifiers [170, 171] or techniques based on Gaussian Copulae [106] and Bayesian networks [14, 172–174].

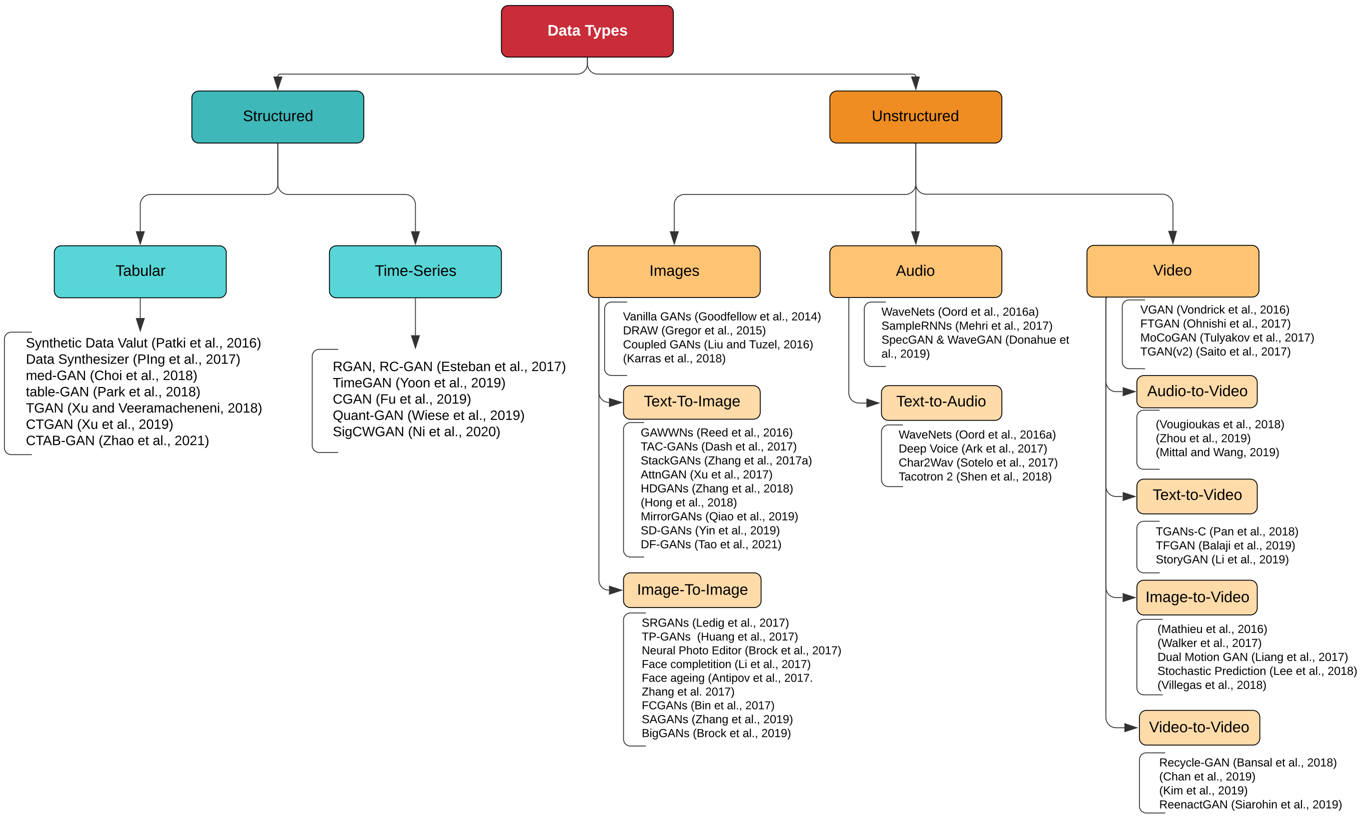
Data Types
Structured
Unstructured
Tabular
Time-Series
Images
Audio
Video
Synthetic Data Valut (Patki et al., 2016)
Data Synthesizer (PIng et al., 2017)
med-GAN (Choi et al., 2018)
table-GAN (Park et al., 2018)
TGAN (Xu and Veeramacheneni, 2018)
CTGAN (Xu et al., 2019)
CTAB-GAN (Zhao et al., 2021)
RGAN, RC-GAN (Esteban et al., 2017)
TimeGAN (Yoon et al., 2019)
CGAN (Fu et al., 2019)
Quant-GAN (Wiese et al., 2019)
SigCWGAN (Ni et al., 2020)
Vanilla GANs (Goodfellow et al., 2014)
DRAW (Gregor et al., 2015)
Coupled GANs (Liu and Tuzel, 2016)
(Karras et al., 2018)
Text-To-Image
GAWWNs (Reed et al., 2016)
TAC-GANs (Dash et al., 2017)
StackGANs (Zhang et al., 2017a)
AttnGAN (Xu et al., 2017)
HDGANs (Zhang et al., 2018)
(Hong et al., 2018)
MirrorGANs (Qiao et al., 2019)
SD-GANs (Yin et al., 2019)
DF-GANs (Tao et al., 2021)
Image-To-Image
SRGANs (Ledig et al., 2017)
TP-GANs (Huang et al., 2017)
Neural Photo Editor (Brock et al., 2017)
Face completition (Li et al., 2017)
Face ageing (Antipov et al., 2017.
Zhang et al. 2017)
FCGANs (Bin et al., 2017)
SAGANs (Zhang et al., 2019)
BigGANs (Brock et al., 2019)
WaveNets (Oord et al., 2016a)
SampleRNNs (Mehri et al., 2017)
SpecGAN & WaveGAN (Donahue et al., 2019)
Text-to-Audio
WaveNets (Oord et al., 2016a)
Deep Voice (Ark et al., 2017)
Char2Wav (Sotelo et al., 2017)
Tacotron 2 (Shen et al., 2018)
VGAN (Vondrick et al., 2016)
FTGAN (Ohnishi et al., 2017)
MoCoGAN (Tulyakov et al., 2017)
TGAN(v2) (Saito et al., 2017)
Audio-to-Video
(Vougioukas et al., 2018)
(Zhou et al., 2019)
(Mittal and Wang, 2019)
Text-to-Video
TGANs-C (Pan et al., 2018)
TFGAN (Balaji et al., 2019)
StoryGAN (Li et al., 2019)
Image-to-Video
(Mathieu et al., 2016)
(Walker et al., 2017)
Dual Motion GAN (Liang et al., 2017)
Stochastic Prediction (Lee et al., 2018)
(Villegas et al., 2018)
Video-to-Video
Recycle-GAN (Bansal et al., 2018)
(Chan et al., 2019)
(Kim et al., 2019)
ReenactGAN (Siarohin et al., 2019)

Figure 2: Generative models for specific data types.

10.1.2 Time-series data

Time-series are series of data points indexed in time order (e.g. electronic health record data containing information about visits to a GP, or higher frequency financial data, such as stock prices). Historically, they were generated via auto-regressive models [175], hardly applicable to practical scenarios where stationarity only holds in specific time regions and in the presence of skewed and heavy-tailed distributions. Conversely, most implicit models focused on conditional distributions of future events given the occurrence of past ones, instead of capturing the full joint-law [16, 99, 176, 177]. Recent developments allow training effort to be optimised by exploiting a reduced feature space, where the data stream is identified by its signature, resulting in a graded sequence of statistics [178].

10.1.3 Images

Applications of image synthesis are extremely diverse, ranging from the reconstruction of a damaged or missing region to the improvement of resolution and colour reproduction. The creation of an image consists in choosing a specific colour for each pixel, as the result of an image-to-image transformation or a text-to-image conversion. Variational Auto-Encoders were somewhat successful [179, 180], however, they featured a pixel-wise loss and simple conditioning, as in the case of class labels and image captions. On the contrary, GANs employ a semantic loss which is more aligned with the human visual system as well as being better suited to highly multi-modal outputs, where several valid images could be created [181]. Starting from vanilla GANs [6], different architectures were proposed to generate plausible images in various datasets, as in the case of human faces [182–188], high-resolution photographs [189–191] and multi-domain images [192, 193]. In text-to-image scenarios, the generation process starts from a brief text description, which is used as additional training information [194, 195] with the eventual aid of stacked architectures [196–198] and attentive, semantic frameworks [199–201] to preserve sentence-level consistency.

10.1.4 Audio

Similar to the generation of time-series, audio signals have a high temporal resolution, requiring representation and synthesis strategies capable of operating efficiently with a large number of dimensions. A significant attempt was the design of WaveNets [202], arising from the architecture of PixelRNN [203] borrowed from the image domain, further evolved by considering the speed difference between the raw audio and the hidden semantic-signal, which is usually many times slower [204, 205]. A different approach consisted in the use of spectrograms, i.e. the simultaneous representations of audio signals in time and frequency, which requires lossy assumptions to cope with their non-invertible nature, inevitably reducing the overall quality [206, 207].

10.1.5 Video

The synthesis of a video can be considered as the generation of a sequence of images, where the main challenge stems from their inter-dependency and hidden temporal dimension. Unconditional video generation tried to maintain scene and foreground consistencies by separately focusing on objects' motion and RGB frame generation [208–211]. Conversely, conditional approaches required smaller training datasets and allowed for finer control of modes of distributions, as in the case of audio conditioning for synchronising speech with a talking character [212–214], text conditioning for video generation [215–217], image conditioning for the prediction of future frames [218–222] and video-to-video for object animation [223–226].

11 Messages from Industry/Start-ups

In preparation of this report, the authors interviewed several industry partners and start-ups in the space of synthetic data. In this section we highlight some of the key themes and messages that we received in response to our questions.

AI itself is in the early stages of mass adoption. Though serious AI research has been ongoing for a long time now, widespread adoption of AI systems is in its infancy. Synthetic data is a younger field than the classical AI/ML problems of prediction, clustering, forecasting, etc. and significant research is required to fully benefit from this technology. That said, there is pressure for the adoption of private synthetic data (more than for other technologies) due to a heightening desire from the public for more privacy control.

Empirical evaluations are necessary. Though differential privacy is an attractive theoretical notion of privacy, industries struggle to trust it without empirical supporting evidence of privacy. Understanding practical implications of differential privacy (i.e. its susceptibility to attacks, what the parameters actually correspond to) is crucial to enable widespread adoption of private synthetic data.

Synthetic data cannot wholly replace real data, or can it? Opinions were more divided on this subject. The impossibility result of Ullman et al. [82] is a blow against the notion of a completely general-use synthetic dataset. The relaxed result of [85], however, indicates it might still be possible in many cases. Several in industry believe that real data should remain at the core of model development, with the final models ultimately being tweaked or even completely re-trained on the real data. Others believe that it will be possible to completely replace real data with synthetic data in the future.

Synthetic data is about enabling. The final takeaway is that synthetic data is about enabling processes that would otherwise not be possible, or that perhaps would drain a lot of resources (such as time). Synthetic data could be used to "access" data across legislative borders (e.g. in companies with an international presence), or to speed up model development times by allowing model designers access to *something* as early as possible. Ultimately, data is very powerful, and synthetic data may allow many more people to tap into its true potential.

12 Conclusion

Synthetic data is a promising technology, with a wide variety of applications. For both privacy and fairness, there is a large cost to getting it wrong. The methods that exist today should be implemented with caution, and significantly more research is needed, from a machine learning perspective, but also from a societal perspective, in order to understand properly the methods that exist.

Acknowledgements

We would like to thank Accenture, Hazy, HSBC, MOSTLY AI, and the Office for National Statistics for their participation in our interviews.

References

[1] James Jordon, Alan Wilson, and Mihaela van der Schaar. Synthetic data: Opening the data floodgates to enable faster, more directed development of machine learning methods. *arXiv preprint arXiv:2012.04580*, 2020.

[2] Samuel Assefa. Generating synthetic data in finance: opportunities, challenges and pitfalls. *Challenges and Pitfalls (June 23, 2020)*, 2020.

[3] Ofer Mendelevitch and Michael D Lesh. Fidelity and privacy of synthetic medical data. *arXiv preprint arXiv:2101.08658*, 2021.

[4] Steven M Bellovin, Preetam K Dutta, and Nathan Reitinger. Privacy and synthetic datasets. *Stan. Tech. L. Rev.*, 22:1, 2019.

[5] Ryan McKenna, Gerome Miklau, and Daniel Sheldon. Winning the nist contest: A scalable and general approach to differentially private synthetic data. *arXiv preprint arXiv:2108.04978*, 2021.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[7] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[8] Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the national academy of sciences*, 99(suppl 3):7280–7287, 2002.

[9] René Carmona and François Delarue. *Probabilistic Theory of Mean Field Games with Applications*, volume 84. Springer, 2018.

[10] Donald B Rubin. *Multiple imputation for survey nonresponse.* New York: Wiley, 1987.

[11] Donald B Rubin. Discussion statistical disclosure limitation. *Journal of official Statistics*, 9(2):461, 1993.

[12] Roderick JA Little. Statistical analysis of masked data. *Journal of Official statistics*, 9(2):407, 1993.

[13] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[14] Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42(4):1–41, 2017.

[15] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018.

[16] Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633*, 2017.

[17] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2018.

[18] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2015.

[19] Faisal Kamiran and Toon Calders. Classifying without discriminating. In *2009 2nd international conference on computer, control and communication*, pages 1–6. IEEE, 2009.

[20] Lu Zhang, Yongkai Wu, and Xintao Wu. A causal framework for discovering and removing direct and indirect discrimination. *arXiv preprint arXiv:1611.07509*, 2016.

[21] Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Optimized pre-processing for discrimination prevention. In *Advances in neural information processing systems*, pages 3992–4001, 2017.

[22] 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.

[23] Boris van Breugel, Trent Kyono, Jeroen Berrevoets, and Mihaela van der Schaar. Decaf: Generating fair synthetic data using causally-aware generative networks. *Advances in Neural Information Processing Systems*, 34, 2021.

[24] Sebastien C Wong, Adam Gatt, Victor Stamatescu, and Mark D McDonnell. Understanding data augmentation for classification: when to warp? In *2016 international conference on digital image computing: techniques and applications (DICTA)*, pages 1–6. IEEE, 2016.

[25] Augustus Odena. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016.

[26] Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Ruslan Salakhutdinov. Good semi-supervised learning that requires a bad gan. *arXiv preprint arXiv:1705.09783*, 2017.

[27] Abhishek Kumar, Prasanna Sattigeri, and Tom Fletcher. Semi-supervised learning with gans: Manifold invariance with improved inference. *Advances in Neural Information Processing Systems*, 30, 2017.

[28] Kasun Bandara, Hansika Hewamalage, Yuan-Hao Liu, Yanfei Kang, and Christoph Bergmeir. Improving the accuracy of global forecasting models using time series data augmentation. *Pattern Recognition*, 120:108148, 2021.

[29] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Data augmentation using synthetic data for time series classification with deep residual networks. *arXiv preprint arXiv:1808.02455*, 2018.

[30] Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pages 233–242. PMLR, 2017.

[31] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[32] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3310–3320, 2017.

[33] Oliver Williams and Frank McSherry. Probabilistic inference and differential privacy. *Advances in Neural Information Processing Systems*, 23: 2451–2459, 2010.

[34] Sahra Ghalebikesabi, Harrison Wilde, Jack Jewson, Arnaud Doucet, Sebastian Vollmer, and Chris Holmes. Bias mitigated learning from differentially private synthetic data: A cautionary tale. *arXiv preprint arXiv:2108.10934*, 2021.

[35] Harrison Wilde, Jack Jewson, Sebastian Vollmer, and Chris Holmes. Foundations of bayesian learning from synthetic data. In *International Conference on Artificial Intelligence and Statistics*, pages 541–549. PMLR, 2021.

[36] Elizabeth J Williamson, Alex J Walker, Krishnan Bhaskaran, Seb Bacon, Chris Bates, Caroline E Morton, Helen J Curtis, Amir Mehrkar, David Evans, Peter Inglesby, et al. Factors associated with covid-19-related death using opensafely. *Nature*, 584(7821):430–436, 2020.

[37] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[38] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 245–248. IEEE, 2013.

[39] 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.

[40] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

[41] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate. *arXiv preprint arXiv:1802.08908*, 2018.

[42] Paul Tiwald, Alexandra Ebert, and Daniel T Soukup. Representative & fair synthetic data. *arXiv preprint arXiv:2104.03007*, 2021.

[43] Sergey I Nikolenko. *Synthetic data for deep learning*. Springer, 2019.

[44] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29 (6):141–142, 2012.

[45] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2017.

[46] Charu C Aggarwal and S Yu Philip. A general survey of privacy-preserving data mining models and algorithms. In *Privacy-preserving data mining*, pages 11–52. Springer, 2008.

[47] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3–es, 2007.

[48] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115. IEEE, 2007.

[49] Mehmet Ercan Nergiz, Maurizio Atzori, and Chris Clifton. Hiding the presence of individuals from shared databases. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 665–676, 2007.

[50] Josep Domingo-Ferrer and Vicenç Torra. A critique of k-anonymity and some of its enhancements. In *2008 Third International Conference on Availability, Reliability and Security*, pages 990–993. IEEE, 2008.

[51] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 111–125. IEEE, 2008.

[52] Hongsheng Hu, Zoran Salcic, Gillian Dobbie, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. *arXiv preprint arXiv:2103.07853*, 2021.

[53] Faiyaz Al Zamal, Wendy Liu, and Derek Ruths. Homophily and latent attribute inference: Inferring latent attributes of twitter users from neighbors. In *Sixth International AAAI Conference on Weblogs and Social Media*, 2012.

[54] Michal Kosinski, David Stillwell, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the national academy of sciences*, 110(15):5802–5805, 2013.

[55] Michal Kosinski, David Stillwell, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the national academy of sciences*, 110(15):5802–5805, 2013.

[56] Mark Bun, Damien Desfontaines, Cynthia Dwork, Moni Naor, Kobbi Nissim, Aaron Roth, Adam Smith, Thomas Steinke, Jonathan Ullman, and Salil Vadhan. Statistical inference is not a privacy violation, 2021. URL `https://differentialprivacy.org/inference-is-not-a-privacy-violation/`.

[57] Frank McSherry. Statistical inference considered harmful, 2016.

URL `https://github.com/frankmcsherry/blog/blob/master/posts/2016-06-14.md`.

[58] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. *arXiv preprint arXiv:1706.08947*, 2017.

[59] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282. IEEE, 2018.

[60] Benjamin Zi Hao Zhao, Aviral Agrawal, Catisha Coburn, Hassan Jameel Asghar, Raghav Bhaskar, Mohamed Ali Kaafar, Darren Webb, and Peter Dickinson. On the (in) feasibility of attribute inference attacks on machine learning models. *arXiv preprint arXiv:2103.07101*, 2021.

[61] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 202–210, 2003.

[62] US Census Bureau. Disclosure avoidance for the 2020 census: An introduction, november 2021. URL `https://www.census.gov/library/publications/2021/decennial/2020-census-disclosure-avoidance-handbook.html`.

[63] Alexandra Wood, Micah Altman, Aaron Bembenek, Mark Bun, Marco Gaboardi, James Honaker, Kobbi Nissim, David R O'Brien, Thomas Steinke, and Salil Vadhan. Differential privacy: A primer for a non-technical audience. *Vand. J. Ent. & Tech. L.*, 21:209, 2018.

[64] Cynthia Dwork. Differential privacy. In *International Colloquium on Automata, Languages, and Programming*, pages 1–12. Springer, 2006.

[65] United States Census Bureau. Census bureau sets key parameters to protect privacy in 2020 census results, 2021. URL `https://www.census.gov/newsroom/press-releases/2021/2020-census-key-parameters.html`.

[66] Microsoft. How statistical noise is protecting your data privacy, 2020. URL `https://news.microsoft.com/on-the-issues/2020/08/27/statistical-noise-data-differential-privacy/`.

[67] Facebook Research. New privacy-protected facebook data for independent research on social media's impact on democracy, 2020. URL `https://research.fb.com/blog/2020/02/new-privacy-protected-facebook-data-for-independent-research-on-social-medias-impact-on-democracy/`.

[68] Frank McSherry and Ilya Mironov. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proceed-*

ings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 627–636, 2009.

[69] Moritz Hardt and Aaron Roth. Beating randomized response on incoherent matrices. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 1255–1268, 2012.

[70] Michael Hay, Chao Li, Gerome Miklau, and David Jensen. Accurate estimation of the degree distribution of private networks. In *2009 Ninth IEEE International Conference on Data Mining*, pages 169–178. IEEE, 2009.

[71] Rui Chen, Benjamin CM Fung, S Yu Philip, and Bipin C Desai. Correlated network data publication via differential privacy. *The VLDB Journal*, 23 (4):653–676, 2014.

[72] Shiva P Kasiviswanathan and Adam Smith. On the'semantics' of differential privacy: A bayesian formulation. *Journal of Privacy and Confidentiality*, 6(1), 2014.

[73] Larry Wasserman and Shuheng Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489): 375–389, 2010.

[74] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275. IEEE, 2017.

[75] Mani Malek Esmaeili, Ilya Mironov, Karthik Prasad, Igor Shilov, and Florian Tramer. Antipodes of label differential privacy: Pate and alibi. *Advances in Neural Information Processing Systems*, 34, 2021.

[76] John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 429–438. IEEE, 2013.

[77] Jaewoo Lee and Chris Clifton. How much is enough? choosing ε for differential privacy. In *International Conference on Information Security*, pages 325–340. Springer, 2011.

[78] Justin Hsu, Marco Gaboardi, Andreas Haeberlen, Sanjeev Khanna, Arjun Narayan, Benjamin C Pierce, and Aaron Roth. Differential privacy: An economic method for choosing epsilon. In *2014 IEEE 27th Computer Security Foundations Symposium*, pages 398–410. IEEE, 2014.

[79] John M Abowd and Ian M Schmutte. An economic analysis of privacy protection and statistical accuracy as social choices. *American Economic Review*, 109(1):171–202, 2019.

[80] Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. Synthetic data–anonymisation groundhog day. *arXiv preprint arXiv:2011.07018*, 2020.

[81] Damien Desfontaines and Balázs Pejó. Sok: differential privacies. *arXiv preprint arXiv:1906.01337*, 2019.

[82] Jonathan Ullman and Salil Vadhan. Pcps and the hardness of generating private synthetic data. In *Theory of Cryptography Conference*, pages 400–416. Springer, 2011.

[83] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. In *Machine learning for healthcare conference*, pages 286–305. PMLR, 2017.

[84] Ahmed M Alaa, Boris van Breugel, Evgeny Saveliev, and Mihaela van der Schaar. How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models. *arXiv preprint arXiv:2102.08921*, 2021.

[85] March Boedihardjo, Thomas Strohmer, and Roman Vershynin. Covariance's loss is privacy's gain: Computationally efficient, private and accurate synthetic data. *arXiv preprint arXiv:2107.05824*, 2021.

[86] Zeyu Ding, Yuxin Wang, Guanhong Wang, Danfeng Zhang, and Daniel Kifer. Detecting violations of differential privacy. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 475–489, 2018.

[87] Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 1895–1912, 2019.

[88] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. Auditing differentially private machine learning: How private is private sgd? *arXiv preprint arXiv:2006.07709*, 2020.

[89] Konstantinos Chatzikokolakis, Tom Chothia, and Apratim Guha. Statistical measurement of information leakage. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 390–404. Springer, 2010.

[90] Giovanni Cherubin. Bayes, not naïve: Security bounds on website fingerprinting defenses. *Proceedings on Privacy Enhancing Technologies*, 4: 135–151, 2017.

[91] Geoffrey Smith. On the foundations of quantitative information flow. In *International Conference on Foundations of Software Science and Computational Structures*, pages 288–302. Springer, 2009.

[92] S Alvim M'rio, Kostas Chatzikokolakis, Catuscia Palamidessi, and Geoffrey Smith. Measuring information leakage using generalized gain functions. In *2012 IEEE 25th Computer Security Foundations Symposium*, pages 265–279. IEEE, 2012.

[93] Giovanni Cherubin, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. F-bleau: fast black-box leakage estimation. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 835–852. IEEE, 2019.

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

[95] MOSTLY AI. Truly anonymous synthetic data - evolving legal definitions and technologies (part ii), 2020. URL `https://mostly.ai/blog/truly-anonymous-synthetic-data-legal-definitions-part-ii/`.

[96] Jinsung Yoon, Lydia N Drumright, and Mihaela Van Der Schaar. Anonymization through data synthesis using generative adversarial networks (ads-gan). *IEEE journal of biomedical and health informatics*, 24 (8):2378–2388, 2020.

[97] Andrew Yale, Saloni Dash, Ritik Dutta, Isabelle Guyon, Adrien Pavao, and Kristin P Bennett. Assessing privacy and quality of synthetic health data. In *Proceedings of the Conference on Artificial Intelligence for Data Discovery and Reuse*, pages 1–4, 2019.

[98] Yunhui Long, Boxin Wang, Zhuolin Yang, Bhavya Kailkhura, Aston Zhang, Carl Gunter, and Bo Li. G-pate: Scalable differentially private data generator via private aggregation of teacher discriminators. *Advances in Neural Information Processing Systems*, 34, 2021.

[99] Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. *Advances in Neural Information Processing Systems*, 32:5508–5518, 2019.

[100] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 343–362, 2020.

[101] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. In *Proceedings on Privacy Enhancing Technologies (PoPETs)*, volume 2019, pages 133–152. De Gruyter, 2019.

[102] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS genetics*, 4(8):e1000167, 2008.

[103] Cynthia Dwork, Adam Smith, Thomas Steinke, Jonathan Ullman, and Salil Vadhan. Robust traceability from trace amounts. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 650–669. IEEE, 2015.

[104] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. Knock knock, who's there? membership inference on aggregate location data. *arXiv preprint arXiv:1708.06145*, 2017.

[105] 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, 2019.

[106] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. The Synthetic Data Vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Montreal, QC, Canada, October 2016. IEEE. ISBN 978-1-5090-5206-6. doi: 10.1109/DSAA.2016.49.

[107] Yuchao Tao, Ryan McKenna, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. Benchmarking differentially private synthetic data generation algorithms. *arXiv preprint arXiv:2112.09238*, 2021.

[108] Claire McKay Bowen and Joshua Snoke. Comparative study of differentially private synthetic data algorithms from the nist pscr differential privacy synthetic data challenge. *Journal of Privacy and Confidentiality*, 11(1), Feb. 2021. doi: 10.29012/jpc.748. URL `https://journalprivacyconfidentiality.org/index.php/jpc/article/view/748`.

[109] James Jordon, Jinsung Yoon, and Mihaela van der Schaar. Measuring the quality of synthetic data for use in competitions. *arXiv preprint arXiv:1806.11345*, 2018.

[110] Mi-Ja Woo, Jerome P Reiter, Anna Oganian, and Alan F Karr. Global measures of data utility for microdata masked for disclosure limitation. *Journal of Privacy and Confidentiality*, 1(1), 2009.

[111] Joshua Snoke, Gillian M Raab, Beata Nowok, Chris Dibben, and Aleksandra Slavkovic. General and specific utility measures for synthetic data. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 181(3):663–688, 2018.

[112] Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. *arXiv preprint arXiv:1806.00035*, 2018.

[113] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *International Conference on Machine Learning*, pages 7176–7185. PMLR, 2020.

[114] Xinyang Zhang, Shouling Ji, and Ting Wang. Differentially private releasing via deep generative model (technical report). *arXiv preprint arXiv:1801.01594*, 2018.

[115] Gergely Acs, Luca Melis, Claude Castelluccia, and Emiliano De Cristofaro. Differentially private mixture of generative neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 31(6):1109–1121, 2018.

[116] Bristena Oprisanu, Georgi Ganev, and Emiliano De Cristofaro. Measuring utility and privacy of synthetic genomic data. *arXiv preprint arXiv:2102.03314*, 2021.

[117] Mayana Pereira, Meghana Kshirsagar, Sumit Mukherjee, Rahul Dodhia, and Juan Lavista Ferres. An analysis of the deployment of models trained on private tabular synthetic data: Unexpected surprises. *arXiv preprint arXiv:2106.10241*, 2021.

[118] Georgi Ganev, Bristena Oprisanu, and Emiliano De Cristofaro. Robin hood and matthew effects–differential privacy has disparate impact on synthetic data. *arXiv preprint arXiv:2109.11429*, 2021.

[119] Victoria Cheng, Vinith M Suriyakumar, Natalie Dullerud, Shalmali Joshi, and Marzyeh Ghassemi. Can you fake it until you make it? impacts of differentially private synthetic data on downstream classification fairness. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 149–160, 2021.

[120] Qingrong Chen, Chong Xiang, Minhui Xue, Bo Li, Nikita Borisov, Dali Kaarfar, and Haojin Zhu. Differentially private data generative models. *arXiv preprint arXiv:1812.02274*, 2018.

[121] Nazmiye Ceren Abay, Yan Zhou, Murat Kantarcioglu, Bhavani Thuraisingham, and Latanya Sweeney. Privacy preserving synthetic data release using deep learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 510–526. Springer, 2018.

[122] Mijung Park, James Foulds, Kamalika Choudhary, and Max Welling. Dp-em: Differentially private expectation maximization. In *Artificial Intelligence and Statistics*, pages 896–904. PMLR, 2017.

[123] Roger B Nelsen. *An introduction to copulas*. Springer Science & Business Media, 2007.

[124] Haoran Li, Li Xiong, and Xiaoqian Jiang. Differentially private synthesization of multi-dimensional data using copula functions. In *Advances in database technology: proceedings. International conference on extending database technology*, volume 2014, page 475. NIH Public Access, 2014.

[125] Zhikun Zhang, Tianhao Wang, Ninghui Li, Jean Honorio, Michael Backes, Shibo He, Jiming Chen, and Yang Zhang. Privsyn: Differentially private data synthesis. In *30th {USENIX} Security Symposium ({USENIX} Security 21)*, 2021.

[126] Donald B Rubin. Multiple imputation after 18+ years. *Journal of the American statistical Association*, 91(434):473–489, 1996.

[127] Jerome P Reiter. Using cart to generate partially synthetic public use microdata. *Journal of Official Statistics*, 21(3):441, 2005.

[128] Gregory Caiola and Jerome P Reiter. Random forests for generating partially synthetic, categorical data. *Trans. Data Priv.*, 3(1):27–42, 2010.

[129] Beata Nowok, Gillian M Raab, and Chris Dibben. synthpop: Bespoke creation of synthetic data in r. *Journal of statistical software*, 74(1):1–26, 2016.

[130] Bronwyn Loong, Alan M Zaslavsky, Yulei He, and David P Harrington. Disclosure control using partially synthetic data for large-scale health surveys, with applications to cancors. *Statistics in medicine*, 32(24):4139–4161, 2013.

[131] John M Abowd and Simon D Woodcock. Disclosure limitation in longitudinal linked data. *Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies*, 215277, 2001.

[132] Jörg Drechsler and Jerome P Reiter. Accounting for intruder uncertainty due to sampling when estimating identification disclosure risks in partially synthetic data. In *International conference on privacy in statistical databases*, pages 227–238. Springer, 2008.

[133] Jerome P Reiter and Robin Mitra. Estimating risks of identification disclosure in partially synthetic data. *Journal of Privacy and Confidentiality*, 1(1), 2009.

[134] Kamalika Chaudhuri and Daniel Hsu. Sample complexity bounds for differentially private learning. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 155–186. JMLR Workshop and Conference Proceedings, 2011.

[135] Jason Tashea. Courts are using AI to sentence criminals. That must stop now., 2017. URL `https://www.wired.com/2017/04/courts-using-ai-sentence-criminals-must-stop-now/`.

[136] Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. Gender bias in neural natural language processing. In *Logic, Language, and Security*, pages 189–202. Springer, 2020.

[137] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018.

[138] Jeffrey Dastin. Amazon scraps secret AI recruiting tool that showed bias against women., 2018. URL `https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G`.

[139] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29:3315–3323, 2016.

[140] Niki Kilbertus, Mateo Rojas-Carulla, Giambattista Parascandolo, Moritz Hardt, Dominik Janzing, and Bernhard Schölkopf. Avoiding discrimination through causal reasoning. *arXiv preprint arXiv:1706.02744*, 2017.

[141] Nina Grgic-Hlaca, Muhammad Bilal Zafar, Krishna P Gummadi, and Adrian Weller. The case for process fairness in learning: Feature selection for fair decision making. In *NIPS symposium on machine learning and the law*, volume 1, page 2, 2016.

[142] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970. PMLR, 2017.

[143] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

[144] Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*, 2020.

[145] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

[146] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[147] Ian Goodfellow. NIPS 2016 Tutorial: Generative Adversarial Networks. *arXiv:1701.00160 [cs]*, 2017.

[148] Brendan J Frey, Geoffrey E Hinton, and Peter Dayan. Does the Wake-sleep Algorithm Produce Good Density Estimators? In *Advances in Neural Information Processing Systems*, volume 8, 1996.

[149] Leo Breiman, Jerome H Friedman, Richard A Olshen, and Charles J Stone. *Classification and regression trees*. Routledge, 2017.

[150] Carl Edward Rasmussen. The infinite gaussian mixture model. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, page 554–560, Cambridge, MA, USA, 1999. MIT Press.

[151] Liangxiao Jiang, Harry Zhang, and Zhihua Cai. A novel bayes model: Hidden naive bayes. *IEEE Transactions on Knowledge and Data Engineering*, 21(10):1361–1371, 2009. doi: 10.1109/TKDE.2008.234.

[152] Ruslan Salakhutdinov and Geoffrey Hinton. Deep Boltzmann Machines. In David van Dyk and Max Welling, editors, *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 448–455, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, April 2009. PMLR.

[153] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 791–798, New York, NY, USA, June 2007. Association for Computing Machinery. ISBN 978-1-59593-793-3. doi: 10.1145/1273496.1273596. URL `https://doi.org/10.1145/1273496.1273596`.

[154] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved Techniques for Training GANs. *arXiv:1606.03498 [cs]*, June 2016.

[155] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv:1701.07875 [cs, stat]*, December 2017.

[156] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. *Advances in neural information processing systems*, 30, 2017.

[157] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

[158] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization. *arXiv:1606.00709 [cs, stat]*, June 2016.

[159] Mario Lucic, Karol Kurach, Marcin Michalski, Olivier Bousquet, and Sylvain Gelly. Are GANs created equal? a large-scale study. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 698–707, Red Hook, NY, USA, December 2018. Curran Associates Inc.

[160] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. *arXiv:1411.1784 [cs, stat]*, November 2014.

[161] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv:1511.06434 [cs]*, January 2016.

[162] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. *arXiv:1606.03657 [cs, stat]*, June 2016.

[163] Yujia Li, Kevin Swersky, and Richard S. Zemel. Generative moment matching networks. *CoRR*, abs/1502.02761, 2015. URL `http://arxiv.org/abs/1502.02761`.

[164] Gregory Caiola and Jerome P. Reiter. Random Forests for Generating Partially Synthetic, Categorical Data. *Trans. Data Priv.*, 2010.

[165] Jörg Drechsler. Using Support Vector Machines for Generating Synthetic Datasets. In Josep Domingo-Ferrer and Emmanouil Magkos, editors, *Privacy in Statistical Databases*, Lecture Notes in Computer Science, pages 148–161, Berlin, Heidelberg, 2010. Springer. ISBN 978-3-642-15838-4. doi: 10.1007/978-3-642-15838-4_14.

[166] Joshua Eno and Craig Thompson. Generating Synthetic Data to Match Data Mining Patterns. *Internet Computing, IEEE*, 12:78–82, June 2008. doi: 10.1109/MIC.2008.55.

[167] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. Generating Multi-label Discrete Patient Records using Generative Adversarial Networks. *arXiv:1703.06490 [cs]*, January 2018.

[168] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling Tabular data using Conditional GAN. *arXiv:1907.00503 [cs, stat]*, October 2019.

[169] Zilong Zhao, Aditya Kunar, Hiek Van der Scheer, Robert Birke, and Lydia Y. Chen. CTAB-GAN: Effective Table Data Synthesizing. *arXiv:2102.08369 [cs]*, May 2021.

[170] Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data Synthesis based on Generative Adversarial Networks. *Proceedings of the VLDB Endowment*, 11(10), June 2018. ISSN 2150-8097. doi: 10.14778/3231751.3231757.

[171] Lei Xu and Kalyan Veeramachaneni. Synthesizing Tabular Data using Generative Adversarial Networks. *arXiv:1811.11264 [cs, stat]*, November 2018.

[172] John M. Abowd and Lars Vilhuber. How Protective Are Synthetic Data? In *Proceedings of the UNESCO Chair in data privacy international conference on Privacy in Statistical Databases*, PSD '08, pages 239–246, Berlin, Heidelberg, September 2008. Springer-Verlag. ISBN 978-3-540-87470-6. doi: 10.1007/978-3-540-87471-3_20.

[173] Haoyue Ping, Julia Stoyanovich, and Bill Howe. DataSynthesizer: Privacy-Preserving Synthetic Datasets. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, Chicago IL USA, June 2017. ACM. ISBN 978-1-4503-5282-6. doi: 10.1145/3085504.3091117.

[174] Grigoriy Gogoshin, Sergio Branciamore, and Andrei S Rodin. Synthetic data generation with probabilistic bayesian networks. *bioRxiv*, 2020. doi: 10.1101/2020.06.14.151084.

[175] Ruey S. Tsay. *Analysis of financial time series.* Wiley, New York, 2002. ISBN 0-471-41544-8 978-0-471-41544-2.

[176] Rao Fu, Jie Chen, Shutian Zeng, Yiping Zhuang, and Agus Sudjianto. Time Series Simulation by Conditional Generative Adversarial Net. *arXiv preprint arXiv:1904.11419*, April 2019.

[177] Magnus Wiese, Robert Knobloch, Ralf Korn, and Peter Kretschmer. Quant GANs: Deep Generation of Financial Time Series. *Quantitative Finance*, July 2019. doi: 10.1080/14697688.2020.1730426.

[178] Hao Ni, Lukasz Szpruch, Magnus Wiese, Shujian Liao, and Baoren Xiao. Conditional Sig-Wasserstein GANs for Time Series Generation. *arXiv:2006.05421 [cs, stat]*, June 2020.

[179] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. DRAW: A Recurrent Neural Network For Image Generation. *arXiv:1502.04623 [cs]*, May 2015.

[180] Elman Mansimov, Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Generating Images from Captions with Attention. *arXiv:1511.02793 [cs]*, February 2016.

[181] Mohamed El-Kaddoury, Abdelhak Mahmoudi, and Mohammed Majid Himmi. Deep Generative Models for Image Generation: A Practical Comparison Between Variational Autoencoders and Generative Adversarial Networks. In Éric Renault, Selma Boumerdassi, Cherkaoui Leghris, and Samia Bouzefrane, editors, *Mobile, Secure, and Programmable Networking*, volume 11557, pages 1–8. Springer International Publishing, Cham, 2019. ISBN 978-3-030-22884-2 978-3-030-22885-9. doi: 10.1007/978-3-030-22885-9_1. URL http://link.springer.com/10.1007/978-3-030-22885-9_1.

[182] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *arXiv:1710.10196 [cs, stat]*, February 2018.

[183] Rui Huang, Shu Zhang, Tianyu Li, and Ran He. Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis. *arXiv:1704.04086 [cs]*, August 2017.

[184] Andrew Brock, Theodore Lim, J. M. Ritchie, and Nick Weston. Neural Photo Editing with Introspective Adversarial Networks. *arXiv:1609.07093 [cs, stat]*, February 2017.

[185] Yijun Li, Sifei Liu, Jimei Yang, and Ming-Hsuan Yang. Generative Face Completion. *arXiv:1704.05838 [cs]*, April 2017.

[186] Grigory Antipov, Moez Baccouche, and Jean-Luc Dugelay. Face aging with conditional generative adversarial networks. In *2017 IEEE International Conference on Image Processing (ICIP)*, September 2017. doi: 10.1109/ICIP.2017.8296650.

[187] Zhifei Zhang, Yang Song, and Hairong Qi. Age Progression/Regression by Conditional Adversarial Autoencoder. *arXiv:1702.08423 [cs]*, March 2017.

[188] Huang Bin, Chen Weihai, Wu Xingming, and Lin Chun-Liang. High-Quality Face Image SR Using Conditional Generative Adversarial Networks. *arXiv:1707.00737 [cs]*, July 2017.

[189] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *arXiv:1609.04802 [cs, stat]*, May 2017.

[190] He Zhang, Vishwanath Sindagi, and Vishal M. Patel. Image De-raining Using a Conditional Generative Adversarial Network. *arXiv:1701.05957 [cs]*, June 2019.

[191] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large Scale GAN Training for High Fidelity Natural Image Synthesis. *arXiv:1809.11096 [cs, stat]*, February 2019.

[192] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised Cross-Domain Image Generation. *arXiv:1611.02200 [cs]*, November 2016.

[193] Ming-Yu Liu and Oncel Tuzel. Coupled Generative Adversarial Networks. *arXiv:1606.07536 [cs]*, September 2016.

[194] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative Adversarial Text to Image Synthesis. *arXiv:1605.05396 [cs]*, June 2016.

[195] Scott Reed, Zeynep Akata, Santosh Mohan, Samuel Tenka, Bernt Schiele, and Honglak Lee. Learning What and Where to Draw. *arXiv:1610.02454 [cs]*, October 2016.

[196] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks. *arXiv:1612.03242 [cs, stat]*, August 2017.

[197] Shikhar Sharma, Dendi Suhubdy, Vincent Michalski, Samira Ebrahimi Kahou, and Yoshua Bengio. ChatPainter: Improving Text to Image Generation using Dialogue. *arXiv:1802.08216 [cs]*, February 2018.

[198] Zizhao Zhang, Yuanpu Xie, and Lin Yang. Photographic Text-to-Image Synthesis with a Hierarchically-nested Adversarial Network. *arXiv:1802.09178 [cs]*, April 2018.

[199] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. AttnGAN: Fine-Grained Text to Image Genera-

tion with Attentional Generative Adversarial Networks. *arXiv:1711.10485 [cs]*, November 2017.

[200] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. MirrorGAN: Learning Text-to-image Generation by Redescription. *arXiv:1903.05854 [cs]*, March 2019.

[201] Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. Inferring Semantic Layout for Hierarchical Text-to-Image Synthesis. *arXiv:1801.05091 [cs]*, July 2018.

[202] 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. *arXiv:1609.03499 [cs]*, September 2016.

[203] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel Recurrent Neural Networks. *arXiv:1601.06759 [cs]*, August 2016.

[204] 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. *arXiv:1612.07837 [cs]*, February 2017.

[205] Jose Sotelo, Soroush Mehri, Kundan Kumar, Joao Felipe Santos, Kyle Kastner, Aaron Courville, and Yoshua Bengio. Char2Wav: End-to-End Speech Synthesis. February 2017.

[206] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, R. J. Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions. *arXiv:1712.05884 [cs]*, February 2018.

[207] Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial Audio Synthesis. *arXiv:1802.04208 [cs]*, February 2019.

[208] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating Videos with Scene Dynamics. *arXiv:1609.02612 [cs]*, October 2016.

[209] Katsunori Ohnishi, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Hierarchical Video Generation from Orthogonal Information: Optical Flow and Texture. *arXiv:1711.09618 [cs]*, December 2017.

[210] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing Motion and Content for Video Generation. *arXiv:1707.04993 [cs]*, December 2017.

[211] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal Generative Adversarial Nets with Singular Value Clipping. *arXiv:1611.06624 [cs]*, August 2017.

[212] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. End-to-End Speech-Driven Facial Animation with Temporal GANs. *arXiv:1805.09313 [cs, eess]*, July 2018.

[213] Hang Zhou, Yu Liu, Ziwei Liu, Ping Luo, and Xiaogang Wang. Talking Face Generation by Adversarially Disentangled Audio-Visual Representation. *arXiv:1807.07860 [cs]*, April 2019.

[214] Gaurav Mittal and Baoyuan Wang. Animating Face using Disentangled Audio Representations. *arXiv:1910.00726 [cs, eess]*, October 2019.

[215] Yingwei Pan, Zhaofan Qiu, Ting Yao, Houqiang Li, and Tao Mei. To Create What You Tell: Generating Videos from Captions. *arXiv:1804.08264 [cs]*, April 2018.

[216] Yogesh Balaji, Martin Renqiang Min, Bing Bai, Rama Chellappa, and Hans Peter Graf. Conditional GAN with discriminative filter generation for text-to-video synthesis. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, pages 1995–2001, Macao, China, August 2019. AAAI Press. ISBN 978-0-9992411-4-1. ZSCC: 0000022.

[217] Yitong Li, Zhe Gan, Yelong Shen, Jingjing Liu, Yu Cheng, Yuexin Wu, Lawrence Carin, David Carlson, and Jianfeng Gao. StoryGAN: A Sequential Conditional GAN for Story Visualization. *arXiv:1812.02784 [cs]*, April 2019.

[218] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv:1511.05440 [cs, stat]*, February 2016.

[219] Alex X. Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic Adversarial Video Prediction. *arXiv:1804.01523 [cs]*, April 2018.

[220] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing Motion and Content for Natural Video Sequence Prediction. *arXiv:1706.08033 [cs]*, January 2018.

[221] Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. The Pose Knows: Video Forecasting by Generating Pose Futures. *arXiv:1705.00053 [cs]*, April 2017.

[222] Xiaodan Liang, Lisa Lee, Wei Dai, and Eric P. Xing. Dual Motion GAN for Future-Flow Embedded Video Prediction. *arXiv:1708.00284 [cs]*, August 2017.

[223] Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. Recycle-GAN: Unsupervised Video Retargeting. *arXiv:1808.05174 [cs]*, August 2018.

[224] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A. Efros. Everybody Dance Now. *arXiv:1808.07371 [cs]*, August 2019.

[225] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep Video Portraits. *arXiv:1805.11714 [cs]*, May 2018.

[226] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating Arbitrary Objects via Deep Motion Transfer. *arXiv:1812.08861 [cs, stat]*, August 2019.