# Generative Adversarial Networks for Sequential Learning

**Tianlin Xu**

Department of Statistics
London School of Economics and Political Sciences

This dissertation is submitted for the degree of
*Doctor of Philosophy*

June 2022

# Declaration

Declaration I certify that the thesis I have presented for examination for the PhD degree of the London School of Economics and Political Science is solely my own work other than where I have clearly indicated that it is the work carried out jointly by me and any other person. The copyright of this thesis rests with the author. Quotation from it is permitted, provided that full acknowledgement is made. This thesis may not be reproduced without my prior written consent. I warrant that this authorisation does not, to the best of my belief, infringe the rights of any third party. I declare that my thesis consists of 24547 words.

I confirm that Chapter 3 was jointly co-authored with Professor Beatrice Acciaio, Dr. Michael Munn and Dr. Wenliang Li, and I contributed 30% of this work. A version of this chapter was published in NeurIPS. Chapter 4 was jointly co-authored with Professor Beatrice Acciaio, and I contributed 70% of this work. A version of this chapter is currently under review. Chapter 5 was jointly co-authored with Professor Beatrice Acciaio, Professor Daniel B Neill and Konstantin Klemmer, and I contributed 30% of this work. A version of this chapter was published in AAAI. Chapter 6 was jointly co-authored with Dr. Chengchun Shi, Dr. Wicher Bergsma and Professor Lexin Li, and I contributed 30% of this work. A version of this chapter was published in the Journal of Machine Learning Research.

Tianlin Xu
June 2022

# Abstract

Generative modelling aims to learn the data generating mechanism from observations without supervision. It is a desirable and natural approach for learning unlabelled data which is easily accessible. *Deep generative models* refer to a class of generative models combined with the usage of deep learning techniques, taking advantage of the intuitive principles of generative models as well as the expressiveness and flexibility of neural networks. The applications of generative modelling include image, audio, and video synthesis, text summarisation and translation, and so on. The methods developed in this thesis particularly emphasise on domains involving data of sequential nature, such as video generation and prediction, weather forecasting, and dynamic 3D reconstruction.

Firstly, we introduce a new adversarial algorithm for training generative models suitable for sequential data. This algorithm is built on the theory of Causal Optimal Transport (COT) which constrains the transport plans to respect the temporal dependencies exhibited in the data. Secondly, the algorithm is extended to learn conditional sequences, that is, how a sequence is likely to evolve given the observation of its past evolution. Meanwhile, we work with the modified empirical measures to guarantee the convergence of the COT distance when the sequences do not overlap at any time step. Thirdly, we show that state-of-the-art results in the complex spatio-temporal modelling using GANs can be further improved by leveraging prior knowledge in the spatial-temporal correlation in the domain of weather forecasting. Finally, we demonstrate how deep generative models can be adopted to address a classical statistical problem of conditional independence testing. A class of classic approaches for such a task requires computing a test statistic using samples drawn from two unknown conditional distributions. We therefore present a double GANs framework to learn two generative models that approximate both conditional distributions. The success of this approach sheds light on how certain challenging statistical problems can benefit from the adequate learning results as well as the efficient sampling procedure of deep generative models.

# Table of contents

# Chapter 1

# Introduction

Given a collection of observations, learning the underlying data distribution has been a central topic in statistics and machine learning. Nevertheless, the purposes of learning can differ. In classic statistics, we learn the data distribution to obtain estimators based on the observed samples in order to make inference about the characteristics of the population. In comparison, generative modelling, which has become an appealing approach in machine learning, learns to generate samples that are as similar as possible to those drawn from the true data distribution. Generative modelling can be a more natural and appropriate approach in the applications where an efficient sampling procedure is prioritised over the understanding of the properties of the learned model parameters. A utilisation of generative models is exemplified by realistic photograph generation, which is particularly useful when the models allow generation conditioned on user inputs, e.g., an image generated based upon a user's description of a scenery or an activity.

Originated in the 1980s, *deep generative models* refer to the class of generative models parametrised using neural networks. As an early attempt, energy-based models [74, 99] started with simple fully connected networks trained to minimise an energy function on the observations, which often associates with a likelihood function. Although theoretically significant, energy-based models are hindered by its inability to scale to high dimensional data as well as its inefficient inference procedure. More recently, more deep generative models such as deep autoregressive models [165, 168, 169], normalising flows [123, 78, 40, 90], variational auto-encoder [91, 120] and generative adversarial networks [67, 118, 104], have been developed to leverage the advances in deep learning and the accessibility of large training datasets. As an unsupervised approach, generative modelling learns through unlabelled training data which is easily accessible. The direct applications of generative modelling include image, audio, and video synthesis, text summarisation and translation, medical imaging, chemical synthesis, and so on. Data synthesis is useful in the scenario where large

datasets are needed for training but unavailable due to privacy concerns or the high cost of data collection.

Deep generative models roughly fall into two categories. *Prescribed generative models* (PGMs) [113] are models whose learning requires the specification of a likelihood function. Most classic models are of the form of PGMs, which follow the conventional lines that allow easy interpretability and straightforward inference. However, the model choice is restricted by the families of likelihood functions that can be written down explicitly. Moreover, the learning outcomes may be compromised to a great extent if a wrong or under-powered class of functions is chosen.

In contrast, *implicit generative models* (IGMs) [44] only specify a data generating procedure without defining a log-likelihood function. In deep generative modelling, IGMs aim to learn a mapping from the latent space to the data space, under the assumption that the high dimensional observations correspond to lower dimensional manifolds which contain unobserved low level features of the observed. In other words, IGMs are typically trained to generate samples that mimic the training data given some latent codes, without the need to estimate the density of them. The learning of IGMs relies on estimating the discrepancy in the empirical distributions of samples from the data distribution and those from the model.

While conceptually attractive, IGMs were long disfavoured due to the lack of efficient learning methods. This has changed by the recent advances in training IGMs in machine learning, driven by the work on generative adversarial networks (GANs) [67]. Since initially proposed in 2014, GANs has become one of the most active area of research. GANs consist of two components: a generator which is an IGM to be learned, and a discriminator which provides information for inference by telling the samples from the model and those from the data distribution apart. In the original GAN, this comparison was completed by a binary classifier that indicates the source of a specific set of samples. The generator and discriminator are then trained simultaneously in an adversarial manner by playing a zero-sum game. This has led to the success of training IGMs that break new ground in terms of sample quality and sampling speed.

Existing literature on GANs can be roughly categorised into three directions. Firstly, a large portion of GAN research contributes to the field by discovering novel network architectures and training techniques from a deep learning perspective. The notorious instability in GAN training caused by the min-max optimisation has motivated this direction of research. To give an example, Karras et al. [84] developed a progressive growing training technique that allows the training process to start with smaller networks for both generator and discriminator and an easier target in which the original images are down-sampled to a lower resolution. As training progresses, more layers are gradually added to the networks

and the resolution of the target images grows higher. In another line of work, the design of generator and discriminator is determined by the domain of applications. For instance, Clark et al. [36] proposed two discriminators in order to capture the spatial and temporal structure of video data separately.

Secondly, another main direction aims to stabilise the training process and maximise model potential by imposing more appropriate regularisers in the training objectives. In generative modelling as well as various areas in machine learning, neural networks are often used as a learned black-box model that approximates a function with constraints. However, due to their complexity and flexibility, it is difficult to restrict them so that certain theoretical requirements can be met by the learned networks. For example, the duality of Wasserstein distance (see details in Chapter 2) computes the difference between two expectations of a function with 1-Lipchitz constraint. To learn this function using neural networks, the initial version of Wasserstein GANs in [8] enforced the constraint on the function space of the discriminator by clipping its parameter values into a fixed interval. This naive implementation can sometimes lead to a failure in generating realistic samples or in the convergence of the loss curve.

Finally, as pointed out in [113], there are a number of approaches that can be used measuring the dissimilarities between the samples from the model and those from the real data distribution, upon which learning of IGMs depends. In GANs, the task of comparing the two sets of samples is fulfilled by the discriminator. In particular, the discriminator in the original GAN algorithm classifies the samples into two classes to indicate which distribution they are drawn from. While this simple binary classifier serves the purpose for adversarial training, the loss values of the GAN objective are not closely associated with the degree of similarity between the generated samples and the real ones. Therefore, researchers placed their attention to the exploration of alternative ways to construct the discriminator so that it provides more meaningful information on how two (empirical) distributions differ. This is the third direction of GAN research.

Unsurprisingly, many classic statistical measures with theoretically superior properties can be adopted for this task. To formulate adversarial training algorithms, existing works along this line typically seek a worst-case comparison, i.e., a maximisation over a parametrised version of a chosen distance, to play the role of the discriminator. Whilst the discriminator learns to maximise a distance to better distinguish the two distributions, the generator parameters are updated to minimise such a distance in order to generate samples whose distribution is as close as possible to the real data distribution. In Chapter 2, we review the variants of GANs derived from these alternative metrics. Our works included in the thesis also fall into this category.

Broadly speaking, statistical approaches compare two distributions using divergences, e.g., Kullback–Leibler (KL) divergence, or metrics, e.g., Max Mean Discrepancy and the Optimal Transport distance. Although widely adopted, the disadvantage of divergences that involve computing a density ratio is well understood. When two distributions have disjoint supports, the divergences from one distribution to another may not be well defined. For this reason, comparison by statistical metrics has grown to be a promising direction of research in the literature of generative modelling. This thesis offers a review of $f$-divergences, Max Mean Discrepancy and Optimal Transport and their applications in generative modelling, with an emphasise on the theory of Optimal Transport due to its close connection to our own works.

Optimal Transport was first posed by French mathematician Gaspard Monge in 1781 [114]. In comparison to divergences, Optimal Transport imposes no restriction on the supports of the distributions between which the distance is of interest. Put simply, the concept of Optimal Transport concerns with the problem of finding a way to move the mass from a starting distribution to a target one with the least cost. This cheapest moving cost can be shown to form a metric over the set of distributions even when the supports of them do not overlap at all. In recent years, this distance has been utilised to construct GAN algorithms, see [8, 37]. For a more rigorous definition and detailed introduction, please refer to Chapter 2.

Sequential data are ubiquitous in the world, including audio and weather recordings, natural languages, physiological and financial traces. In comparison, sequential data tracks a sample over time and exhibits different characteristics from static data such as images. A gap to be filled in the GAN literature is the absence of an appropriate distance measure for learning sequential data. The majority of existing works in sequential generative modelling merely rely on certain specific network architectures, e.g., Recurrent Neural Networks (RNNs) or 1D Convolution Neural Networks (CNNs), to capture the sequential nature of the data.

Nevertheless, when comparing the difference between two distributions on sequences, the aforementioned statistical metrics are often applied to conduct a point-wise comparison, i.e., treating the observations along the time dimension as if they are independent of the past and future. This is not ideal. The inconsistency between a network design that respects the temporal dependencies in the data and an objective function that ignores it may lead to a compromise in model performance, especially if we want to give importance to apprehending the evolution of sequences over time.

This issue persists in the modelling of data characterised by spatio-temporal complexities, which has a range of impactful downstream applications such as video generation, weather forecasting, and dynamic 3D reconstruction. Training IGMs on spatio-temporal dynamics

poses a difficult challenge. On one hand, learning complex spatial structures of static images has already received significant effort within the research community. On the other hand, temporal dependencies are no less complicated since the dynamical features are strongly correlated with spatial features. The success of a generative model is measured based upon its ability to not only generate high-quality images at every time step but also understand the evolution of motions in the images over the course of time.

Similarly, most GANs tackle spatio-temporal learning by using specialised network architectures. A combination of Convolution Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are often deployed to capture the spatial and temporal features exhibited in the training data separately, see e.g. [151, 4, 136, 172, 162]. Alternatively, Convolutional LSTM [145] are also considered to be a more compact model, accompanied by a higher computational cost.

Although these network structures have been proved successful to a certain degree in learning the underlying dynamics of spatio-temporal data, an improvement may be achieved if the training objective can also take the characteristics of the data into account. Ultimately, the parameter learning in IGMs is guided by the information acquired from the comparison between the generated and the real samples via an objective function. In fact, some attempted to develop objective functions more suitable for sequential learning by dividing them into static and dynamic components, see e.g. [190, 46]. However, this type of objective functions are usually designed for empirical experiments in a specific domain. They are less generic and lack theoretical justification and guarantees such as convergence, unlike the classic statistical metrics.

All our works in this thesis are devoted to examining the adversarial algorithms for training dynamic IGMs for sequential data. We first introduce a new adversarial objective that builds on the theory of Causal Optimal Transport which constrains the transport plans to respect *causality*: the probability mass moved to the target sequence at time $t$ can only depend on the source sequence up to time $t$, see [2, 12]. In this way, at every time we only use information available up to that time, Causal Optimal Transport provides a natural distance that compares how two processes evolves differently over time. A reformulation of the causality constraint leads to a new adversarial training objective tailored to sequential data. This is the foundation of COT-GAN, introduced in Chapter 3 (see also [186]). The success of the algorithm also relies on a new, improved version of the Sinkhorn divergence [62] which demonstrates less bias in learning.

In Chapter 4 (see also [185]), COT-GAN is extended to learn conditional sequences, that is, how a sequence is likely to evolve given the observation of its past evolution. Meanwhile, we address the issue that the Causal Optimal Transport distance between a distribution and

the empirical measure of a sample from it may not vanish while the size of the sample goes to infinity. We thereby proposed KCCOT-GAN (Kernel Conditional COT-GAN) which employs a modification of the empirical measures via kernel smoothing [127] in order to yield better convergence properties.

In Chapter 5 (see also [95]), we show that state-of-the-art results in the complex spatio-temporal modelling can be further improved by leveraging human expert knowledge in specific domains such as weather forecasting. Spatio-temporal autocorrelations based upon human understanding can be then encoded as embedding loss into COT-GAN to formulate a new GAN framework, named SPATE-GAN. We test this new objective on a diverse set of complex spatio-temporal patterns: turbulent flows, log-Gaussian Cox processes and global weather data. We show that this novel embedding loss improves performance without any changes to the architecture of the COT-GAN backbone, highlighting the increased model capacity for capturing the spatial-temporal structures in the training data.

Beyond the direct applications, conditional independence testing can also benefit from learning about the data through generative modelling. Testing conditional independence is a key building block and plays a central role in a wide variety of statistical learning problems, for instance, causal inference [125], graphical models [96], dimension reduction [102], among others. The key question to answer in conditional independence testing is whether two random variables $X$ and $Y$ are conditionally independent given a set of confounding variables $Z$. A class of approaches for such a task requires computing a test statistic using samples drawn from the unknown conditional distributions $P(X|Z)$ and $P(Y|Z)$.

In Chapter 6 (see also [144]), we introduce a double GANs framework to learn two generators of the conditional distributions for the problem of high-dimensional conditional independence testing. We then integrate the two generators to construct a test statistic that is doubly robust, and can both control type-I error and has the power approaching one asymptotically. Also notably, we establish those theoretical guarantees under much weaker and practically more feasible conditions compared to the existing tests. This gives a concrete example of how to utilise the state-of-the-art machine learning models, such as GANs, to help address a classical but challenging statistical problem.

# Chapter 2

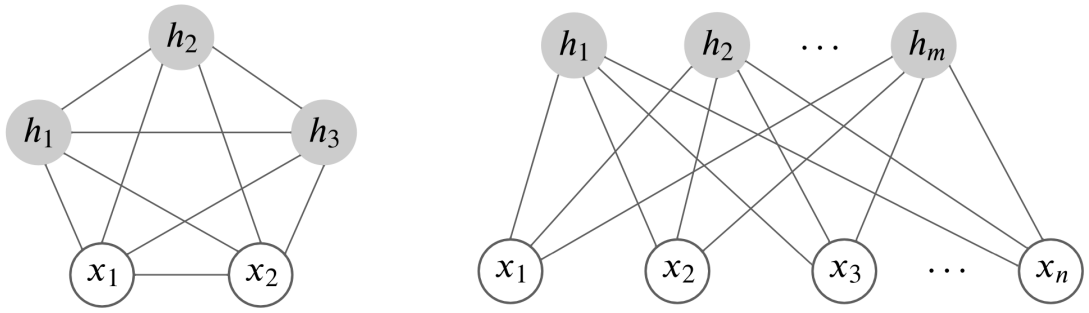
# Foundations

In this chapter, we introduce classic deep generative models and the fundamental concepts of Optimal Transport and Causal Optimal Transport which are the theoretical foundation for the remainder of the thesis. In particular, we revise the most representative classes of generative models from the perspective of probabilistic modelling. An emphasis is placed on a subset of generative models, namely Generative Adversarial Networks, especially those derived from the theory of Optimal Transport.

## 2.1 Deep generative models

Deep generative modelling includes a class of techniques that aim to learn the underlying data generating process from observations using neural networks without supervision. Typically, we assume that the high dimensional observations correspond to lower dimensional manifolds which contains unobserved low level features of the data. This can be represented using latent variables which are assumed to exist but not observed. To avoid confusion, we denote the latent variables that are sampled from a distribution on some (low-dimensional) latent space $\mathcal{Z}$ as $z = \{z_i\}_{i=1}^{M}$, whereas the learned latent variables in the model, also known as hidden units in neural networks, are denoted as $h = \{h_i\}_{i=1}^{J}$ for $h_i \in \mathcal{X}$.

In Section 2.1, we briefly review classic prescribed generative models (PGMs) including energy-based models, autoregressive models, normalising flows as well as variational auto-encoders. Readers who are familiar with classic PGMs in the context of generative modelling can safely skip this section. In Section 2.2, we discuss several variants of Generative Adversarial Networks, and introduce the concepts of Optimal Transport and Causal Optimal Transport.




(a) Boltzmann Machines (b) Restricted Boltzmann Machines

Fig. 2.1 Diagrams for Boltzmann Machines and Restricted Boltzmann Machines.

### 2.1.1 Energy-based models

Given $x \in \mathbb{R}^D$, an energy function $E : \mathbb{R}^D \mapsto \mathbb{R}$ associates high probability of a system being in state $x$ with low energy and low probability of a system being in state $x$ with high energy. Examples of functions of this nature include contrast functions, value functions, or negative likelihood functions. Stemming from statistical mechanics, most energy-based generative models eventually convert the energies for possible outcomes into a normalised probability distribution through the Gibbs distribution (also called Boltzmann distribution),

$$p(x) = \frac{e^{-\beta E(x)}}{\int_{x' \in \mathcal{X}} e^{-\beta E(x')}}, \tag{2.1.1}$$

where $\beta$ is an arbitrary positive constant representing the inverse temperature of the system. The Gibbs distribution can describe a large class of distributions determined by a specific choice of the energy function, provided that the integral in the denominator is well defined.

#### Boltzmann machines and Restricted Boltzmann machines

A Boltzmann machine is a class of energy-based model that utilises a fully connected undirected network of binary neurons, see Figure 2.1 (a) for an illustration. The state of a neuron is determined by a weighted sum of all neurons. Given all variables $s = (x,h)$ where $x = \{x_i\}_{i=1}^N$ are binary observations for $x_i \in \{0,1\}^D$ (also called visible variables) and $h = \{h_i\}_{i=1}^J$ are hidden variables for $h_i \in \{0,1\}^M$, the probability of state $s_i$ is given by

$$p(s_i = 1) = \sigma(\sum_j w_{i,j} s_j) \tag{2.1.2}$$

where $\sigma(.)$ is the sigmoid function and $w_{i,j}$ is the weight at $(i,j)$ position of the learned weight matrix $W$. The resulting neuron can be seen as either on or off. It is known that Boltzmann machines with only visible variables lack the capacity to capture the characteristics of the data distribution beyond its first and second moments. Latent variables are therefore introduced in order to increase the model capacity. Boltzmann machine chooses an energy function that allows full connections between the visible and latent variables,

$$E(x,h) = -\frac{1}{2}x^\top Wx - \frac{1}{2}h^\top Jh - \frac{1}{2}x^\top Lh, \tag{2.1.3}$$

where $W$, $J$ and $L$ are symmetrical learned weight matrices.

As the latent variables are unobserved, the inference of the parameters involves integrating them out. Due to the binary nature of the variables, the probability of visible variables can be written as finite summations,

$$p(x) = \frac{\sum_h e^{-E(x,h)}}{\sum_{x,h} e^{-E(x,h)}}. \tag{2.1.4}$$

To facilitate inference and parameter learning, one way is to constrain off-diagonal entries in the weight matrices to zero, which can be seen as a restriction of connectivity. This leads to the restricted Boltzmann machines which forbids communications between the variables in the same layer, see Figure 2.1 (b). This approach simply sets $J = 0$ and $L = 0$ in Equation (2.1.3).

Denoting the parameters by $\theta := (W,J,L)$ and the model by $p_\theta(s)$ where $s = (x,h)$ as aforementioned, we can find the gradient of $\log p_\theta(s)$ with respect to $W$,

$$\sum_s \frac{\partial \log p_\theta(s)}{\partial w_{i,j}} = \mathbb{E}_x[xh^\top] - \mathbb{E}_h[xh^\top]. \tag{2.1.5}$$

The expectations in Equation (2.1.5) is called contrastive divergence which is intractable in practice as the computational time required grows exponentially with the number of hidden units in the model. As a result, an approximation is typically obtained by sampling the observed and hidden units using a Gibbs sampler during both the training and inference stages. This slow iterative process largely hinders the applicability of Boltzmann machines in high dimensional data, making it impossible to be adopted for large-scale machine learning tasks.

### 2.1.2 Autoregressive models

This class of generative models learns the data distribution directly in an autoregressive manner. According to the chain rule of probability, autoregressive models predict the

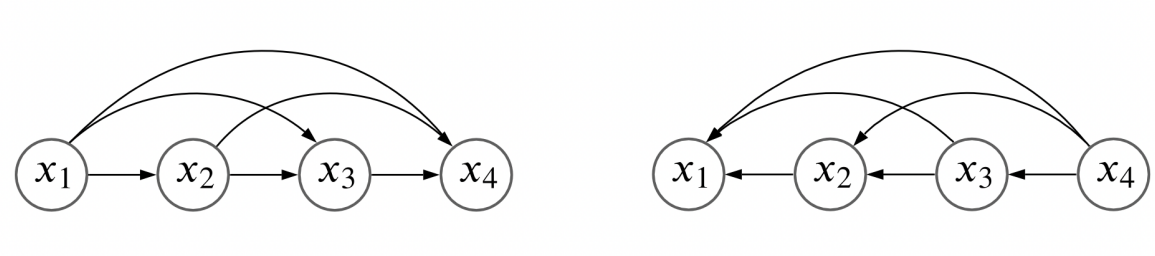



Fig. 2.2 Diagrams showing two different autoregressive models defined by different orderings of the variables.

conditional distribution over visible variables one dimension at a time, following a particular order. An observed variable of dimension $D$, i.e., $x \in \mathbb{R}^D$, can be decomposed as $x^1, ..., x^D$ over its dimensions. The probability of $x$ can be expressed as as a product of a series of conditional probabilities,

$$p(x) = \prod_{d=1}^{D} p(x^d | x^1, ..., x^{d-1}). \tag{2.1.6}$$

The ordering of decomposition is to be determined, usually according to the nature of the observations. For example, for a 1D time series $x^1, ..., x^D$ over $D$ time steps, the decomposition is typically chosen to respect the fact that the data is generated in an autoregressive manner over time. See Figure 2.2 for illustrations for autoregressive models defined by two different orderings of the variables.

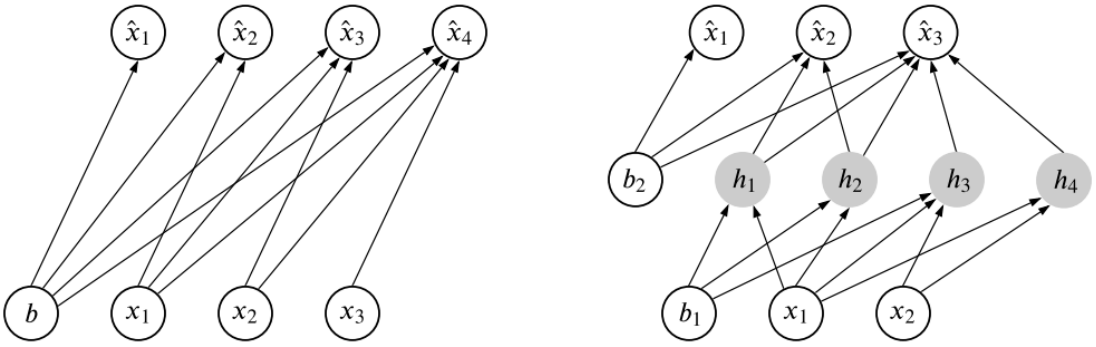
It is possible to directly maximise the likelihood given the observed variable $x$ by minimising the negative log-likelihood,

$$-\log p(x) = -\sum_{d=1}^{D} \log p(x^d | x^1, ..., x^{d-1}). \tag{2.1.7}$$

**Autoregressive neural network models**

Frey [55] and Bengio and Bengio [20] explored implementations of the autoregressive conditionals in Equation (2.1.6) using neural networks. Frey [55] first implemented a multidimensional binary variable with autoregressive connections by using logistic regression to model each conditional distribution,

$$p(x^d = 1 | x^{\leq d-1}) = \sigma(W^{(d)} x^{\leq d-1} + b_d) \tag{2.1.8}$$




(a) Logistic autoregressive Bayesian networks models

(b) Autoregressive neural network models

Fig. 2.3 Diagrams for Logistic autoregressive Bayesian networks and Autoregressive neural network models.

where $x^{\leq d-1} = (x^1, ..., x^{d-1})$, $\sigma(\cdot)$ is the sigmoid function, the parameters $W^{(d)} \in \mathbb{R}^{d-1}$, and $b_d$ is the bias that belongs to $\mathbb{R}$. The diagram for logistic autoregressive Bayesian networks (LABNs) is shown in Figure 2.3 (a).

To improve the model capacity, Bengio and Bengio [20] extended this model to include hidden units in learning the autoregressive conditionals. The key idea is to restrict the model to only allow $k$ causal connections between the variables from connecting layers. For $k = 2$, the variables in subsequent layers can only access two units from the past in the previous layer, see Figure 2.3 (b) for an illustration for autoregressive neural network models with causal connections with $k = 2$.

Formally, for hidden variables $h_i \in \{0,1\}^M$, the distribution over the $d^{th}$ dimension of $x$ is calculated as the following:

$$p(x^d = 1 | x^{\leq d-1}) = \sigma\left(\sum_{j=1}^{k(d-1)} W^{d,j} h_j + b_d^v\right), \tag{2.1.9}$$

$$h_j = \tanh\left(\sum_{d=1}^{j//k+1} V^{j,d} x_i^d + b_j^h\right), \tag{2.1.10}$$

where $W$ and $V$ are weight matrices, in which many entries are zero due to the restricted connectivity. The autoregressive neural network models reported excellent statistical performance [140, 169], as measured by the model likelihood on a test dataset. They also found it advantageous to reduce the number of free parameters by further pruning the connectivity between the inputs and hidden units.

### Neural autoregressive distribution estimators

The Neural autoregressive distribution estimator (NADE) [165] belongs to the class of autoregressive neural network models, which also relied on binary units in a neural network with a single hidden layer. NADE adopted the mean-field variational approximation (see [26]) to the conditionals in Equation (2.1.6), each of which was modelled by a binary restricted Boltzmann machine. Thus, the conditional distribution is defined as a restricted Boltzmann machine with the $m$ hidden variables connected to the $d^{th}$ dimension of the visible variable $x \in \{0,1\}^D$:

$$p(x^d|x^{\leq d-1}) = \frac{p(x^d, x^{\leq d-1})}{p(x^{\leq d-1})} = \frac{\sum_{k=d+1}^{D} \sum_h e^{-E(x^k,h)}}{\sum_{k=d}^{D} \sum_h e^{-E(x^k,h)}}. \tag{2.1.11}$$

where we abuse the notation $h$ to refer to the sets of hidden variables connected to the corresponding visible variables involved in the numerator and denominator.

Unfortunately, most conditionals for the dimensions of visible variables are intractable in practice as the normalising constant in the denominator requires a summation over an exponential number of configurations of the visible variables. Uria et al. [165] opted for the mean-field variational approximation (see Blei et al. [26]) to the joint distribution $p(x^{\geq d}, h|x^{<d})$. The factorial nature of the mean-field variational approximation makes it easy for the marginalisation of hidden variables. Furthermore, the variational distribution $q(x^{\geq d}, h|x^{<d})$ that approximates the joint distribution $p(x^{\geq d}, h|x^{<d})$ is chosen to be a product of two Bernoulli distributions due to the binary nature of the variables. To learning the parameters of the variational distribution, NADE minimises the KL-divergence between $q(x^{\geq d}, h|x^{<d})$ and $p(x^{\geq d}, h|x^{<d})$, from which the analytical forms of parameter updates can be derived. Uria et al. [166] proposed real-valued NADE as an extension to model the joint distribution $p(x^{\geq d}, h|x^{<d})$ for real-valued visible variables and binary hidden variables.

Later developments in autoregressive models, such as PixelRNN [169], Wavenet [168], and PixelCNN++ [140], are proved to be promising in multiple application areas, namely image, audio, and language generation and prediction. Although powerful and straightforward to train, the disadvantages of autoregressive models are widely understood, including compounding errors and inefficiency in sampling due to the sequential nature of its generating process, see e.g., [82, 130]. A modelling challenge lies in the determination of the ordering of decomposition based upon the dependencies between variables. Whilst the ordering is apparent for some data such as audio and text, it is not for others such as images or videos. This, however, can be crucial for the model performance.

### 2.1.3 Flow-based models

Similar to autoregressive models, flow-based models, also known as normalising flows, are a class of prescribed generative models that maximise a log-likelihood function.

Recall the result of change of variables in the probability density function: given a random variable $z \sim q(z)$ and an invertible, smooth function $g : \mathbb{R}^D \mapsto \mathbb{R}^D$, the transformed random variable $x = g(z)$. The distribution of $x$ can be determined via the change of variables rule,

$$p(x) = q(z)\left|\det\frac{\partial g^{-1}}{\partial x}\right| = q(z)\left|\det\frac{\partial g}{\partial z}\right|^{-1}. \tag{2.1.12}$$

where $g^{-1}$ is the inverse function of $g$.

Flow-based models cleverly make use of this transformation from one variable to another. In the context of generative modelling, the random variable $z$ plays the role of the latent component, and the function $g$ is the generative model that maps the latent space $\mathcal{Z}$ to the data space $\mathcal{X}$. The generative model is then trained to capture the characteristics of the underlying data distribution.

Based upon this principle, the complexity of flow models can be built up by a series of compositions of transformations from the original latent variable $z$. The transformed random variable $x$ after $K$ transformations is therefore defined by

$$x = g_K \circ \cdots \circ g_2 \circ g_1(z), \tag{2.1.13}$$

where the composition of $K$ functions, denoted as $g(z) = g_K \circ \cdots \circ g_2 \circ g_1(z)$, is the generative model that produces data $x$ given some noise in the latent space.

The log likelihood function of the resulting distribution through this chain of transformations can be found by applying the change of variables rule in Equation (2.1.12),

$$\log p(x) = \log q(z) - \sum_{k=1}^{K} \log\left|\det\frac{\partial g_k}{\partial z_{k-1}}\right|. \tag{2.1.14}$$

Note that Equation (2.1.12) requires the generative model $g$ to be invertible. To exploit the power of deep learning techniques, many flow-based models focus on exploring the options of constructing expressive invertible functions with neural networks. A coupling flow [45] is a method that divides input data into two blocks and applies a bijection transformation on one of the blocks.

Let $x \in \mathbb{R}^D$, $x_{I_1}$ and $x_{I_2}$ partitions of $x$ such that $x_{I_2} \in \mathbb{R}^d$ and $x_{I_2} \in \mathbb{R}^{D-d}$ for $d \in \{1,2,...,D-1\}$, and $h$ a function $h : \mathbb{R}^d \mapsto \mathbb{R}^d$, we can define $y = (y_{I_1}, y_{I_2})$ as

$$\begin{aligned} y_{I_1} &= x_{I_1}, \\ y_{I_2} &= g(x_{I_2}, h(x_{I_1})), \end{aligned}$$

where $g : \mathbb{R}^{D-d} \times \mathbb{R}^d \mapsto \mathbb{R}^{D-d}$ is an invertible map with respect to its first argument given the second, and $h$ can be an arbitrary function, e.g., a neural network. Coupling flow improves the computational efficiency of the Jacobian determinant in Equation (2.1.14) by choosing an element-wise function $g$. To see this, we write the Jacobian matrix of $y$:

$$\frac{\partial y}{\partial x} = \begin{bmatrix} I_d & 0 \\ \frac{\partial y_{I_2}}{\partial x_{I_1}} & \frac{\partial y_{I_2}}{\partial x_{I_2}} \end{bmatrix}, \tag{2.1.15}$$

where $I_d$ is the identity matrix of size $d$. The $\det \frac{\partial y}{\partial x}$ is simply $\det \frac{\partial y_{I_2}}{\partial x_{I_2}}$ whose computation sorely depends on the choice of function $g$.

Papamakarios et al. [123] introduced autoregressive flows which can be seen as a more flexible generalisation of coupling flow by choosing a dynamic partition. Alternative flows are also proposed, see [78, 40, 90]. In general, flow-based models are less efficient because of the restriction on function invertibility as well as the computation of the determinant of the Jacobian matrix.

### 2.1.4 Variational auto-encoders

Variational auto-encoders (VAEs) are a class of generative models that maximise a lower bound on a log-likelihood function, derived from the study of variational inference (see [26]). As latent variable models, the inference of VAEs also requires integrating out the latent variables. In another word, the learning relies on the computing the following integral,

$$p(x) = \int p(x,z)dz = \int p(x|z)p(z)dz. \tag{2.1.16}$$

However, the integral is only tractable for extremely limited choices of generative models and prior distributions $p(z)$ (see probabilistic principle component analysis [24] for an example). In the cases of this integral being intractable, we can approximate the log of $p(x)$ using variational inference which obtains an approximation to the exact posterior distribution:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}. \tag{2.1.17}$$

For a variational distribution $q(z|x) \in \mathcal{Q}$, where $\mathcal{Q}$ is a family of distributions, the key idea of variational inference is to learn $q(z|x)$ to approximate the posterior $p(z|x)$, thereby turning the inference problem into an optimisation problem. More specifically, the Kullback–Leibler (KL) divergence from $q(z|x)$ to $p(z|x)$ is minimised when an optimal solution ($q(z|x) = p(z|x)$) is found,

$$\arg\min_{q(z|x)\in\mathcal{Q}} D_{KL}(q(z|x)||p(z|x)), \tag{2.1.18}$$

where $D_{KL}(\cdot||\cdot)$ denotes the KL divergence, which can be expanded and rearranged into

$$D_{KL}(q(z|x)||p(z|x)) = \mathbb{E}_{q(z|x)}[\log q(z|x)] - \mathbb{E}_{q(z|x)}[\log p(z,x)] + \log p(x). \tag{2.1.19}$$

Due to the fact that $D_{KL}(q||p) \geq 0$ for any two distributions $p$ and $q$, we can set the right hand side of Equation (2.1.19) to be greater than zero and rearrange the terms to obtain the evidence lower bound (ELBO) on $\log p(x)$,

$$\log p(x) \geq -\mathbb{E}_{q(z|x)}[\log q(z|x)] + \mathbb{E}_{q(z|x)}[\log p(z,x)] \tag{2.1.20}$$

$$= \mathbb{E}_{q(z|x)}[\log p(x|z)] - D_{KL}(q(z|x)||p(z)). \tag{2.1.21}$$

We proceed to explain how the ELBO can be used to train a generative model. VAEs sample the latent variable $z$ from a prior distribution $p(z)$ and generate $x$ from the conditional distribution $p(x|z)$. The conditional model mapping from the latent space to the data space is called the *decoder*. In addition, VAEs also propose the variational model $q(z|x)$ to be an *encoder* which maps the observations into latent features. Both the encoder and decoder are trained by maximising the ELBO on the log-likelihood function of $p(x)$.

### 2.1.5 Diffusion models

Diffusion probabilistic generative models (see e.g. [149, 76]) are latent variable models in which the latent variables $x_1, ..., x_T$ where $x_t \in \mathbb{R}^D$ for all $t = 1, ..., T$ are defined as a Markov chain. In the forward (diffusion) process, the original data $x_0 \in \mathbb{R}^D$ is gradually corrupted through $T$ diffusion steps by $x_1, ..., x_T$. The key idea is that, if the corrupted data distribution at diffusion step $T$ tends to a noise distribution which permits efficient sampling, we can train a generative model that learns the reverse process to convert the noise to the original data.

Consider a Markov process with Gaussian transitions. The forward process can be written as

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}), \quad q(x_t|x_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t \mathbf{I}), \tag{2.1.22}$$

where $\beta_t$ is the variance for the $t^{\text{th}}$ diffusion step.

Note that the product of the Gaussian distributions up to step $t$ in the forward process is equivalent to

$$q(x_t|x_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{2.1.23}$$

where $\alpha_t = 1 - \beta_t$, and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. This formulation admits efficient transformation from $x_0$ to $x_t$ at an arbitrary step $t$.

The generative model, parameterised by $\theta$, is defined as the reverse process that gradually denoises the noisy inputs back to the original data:

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \quad p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t,t), \Sigma_\theta(x_t,t)), \tag{2.1.24}$$

where the starting noise distribution is $p(x_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$.

From the perspective of variational inference, the forward process can be viewed as the variational distribution that approximates the posterior distribution of the latent variables in the reverse process, i.e., $p_\theta(x_{1:T}|x_0)$. This leads to a training objective that minimises the evidence lower bound (ELBO) on the negative log-likelihood:

$$\mathbb{E}[-\log p_\theta(x_0)] \leq \mathbb{E}_q \left[ \log p(x_T) - \sum_{t \geq 1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} \right]. \tag{2.1.25}$$

Sohl-Dickstein et al. [149] showed that the ELBO (2.1.25) can be rewritten as a variance reduced t-step comparison between the reverse process and the posteriors of the forward process:

$$\mathbb{E}_q \left[ D_{KL}(q(x_T|x_0)||p(x_T)) + \sum_{t>1} D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t)) - \log p_\theta(x_0|x_1) \right], \tag{2.1.26}$$

where $q(x_{t-1}|x_t, x_0)$ is the forward process posterior conditioned on $x_0$ with mean $\tilde{\mu}_t$ and variance $\tilde{\beta}_t$:

$$\tilde{\mu}_t(x_t, x_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t \quad \text{and} \quad \tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \tag{2.1.27}$$

The KL divergence between two Gaussian distributions come down to mean and variance matching. By choosing the variance of the reverse process $\Sigma_\theta(x_t, t) = \sigma^2\mathbf{I}$, the training loss (2.1.26) at step $t$ for all $t = 1, ..., T-1$ is reduced to

$$\mathbb{E}_q\left[\frac{1}{2\sigma^2}||\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)||^2\right] + C, \tag{2.1.28}$$

where $C$ is a constant.

Whilst it is possible to parametrise a model to directly learn the forward posterior mean [149], Ho et al. [76] opted for a different parameterisation that learns the noise schedule in the forward process. Recall that we can obtain $x_t$ at any arbitrary step by $x_t(x_0, \varepsilon) = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon$ for $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ from Equation (2.1.23). We then write $x_0$ as a function of $x_t$, substitute it in the posterior mean (2.1.27), and yield an alternative expression:

$$\tilde{\mu}_t(x_t, x_0) = \frac{1}{\sqrt{\alpha_t}}\big(x_t(x_0, \varepsilon) - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon\big). \tag{2.1.29}$$

By parametrising the model as $\tilde{\mu}_\theta(x_t, x_0) = \frac{1}{\sqrt{\alpha_t}}\left(x_t(x_0, \varepsilon) - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta\right)$, Ho et al. [76] arrived at a simplified objective that matches the noise at diffusion step $t$ for $t = 1, ..., T-1$:

$$\mathbb{E}_{x_0, \varepsilon, t}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}||\varepsilon - \varepsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon, t)||\right]. \tag{2.1.30}$$

The sampling procedure of diffusion models is known to be inefficient due to its iterative nature. A sequence $x_T, ..., x_0$ is sampled until the original $x_0$ is recovered. Starting from a prior distribution $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we obtain $x_{t-1}$ iteratively by

$$x_{t-1} = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}\varepsilon_\theta(x_t, t)}\right) + \sigma_t z \tag{2.1.31}$$

where $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The sampling procedure coincides with the Langevin dynamics if we view the learned noise $\varepsilon_\theta$ as the gradient of data density. The slow sampling speed is one of the major obstacles of diffusion models to be applied at large scale. Recent efforts have focused on improving the sampling efficiency [32, 179, 139].

## 2.2 Generative adversarial networks

As an extension to generative stochastic networks [5**?** ], generative adversarial networks (GANs) [67] is a new class of training scheme that allows training an implicit generative model (IGM) without explicitly specifying a representation of the likelihood function. Before introducing a number of variants of GANs, we formally define the task of IGMs training: given a (real) data distribution $\mu = \frac{1}{N}\sum_{i=1}^{N} \delta_{x^i}$, $x^i \in \mathcal{X}$, and a distribution $\zeta$ on some latent space $\mathcal{Z}$, the generator is a function $g : \mathcal{Z} \to \mathcal{X}$ trained so that the induced distribution $\nu = \zeta \circ g^{-1}$ is as close as possible to $\mu$.

### 2.2.1 Original GAN

Initially proposed in 2014 [67], GANs introduced an training scheme which consists of two components, a discriminator and a generator. While the generator learns to produces plausible samples, the discriminator provides information on how the two distributions differ. What's special about GANs is that its generator and discriminator are trained simultaneously in an adversarial manner by playing a zero-sum game.

In the original GAN, the discriminator is a function $f : \mathcal{X} \to [0,1]$ trained to output a high value if the input is real (from $\mu$), and a low value otherwise (from $\nu$). Both the generator and discriminator are parametrised as neural networks with parameters $\theta$ and $\varphi$, denoted as $g_\theta$ and $f_\varphi$ respectively. The objective function of the original GANs is formulated as a min-max optimisation over a binary classification loss:

$$\min_{\theta}\max_{\varphi} \mathbb{E}_{x\sim\mu}[\log f_\varphi(x)] + \mathbb{E}_{y\sim\nu}[\log(1 - f_\varphi(g_\theta(z)))]. \tag{2.2.1}$$

The objective function is maximised over the discriminator parameter $\varphi$ to learn a better classifier, and minimised over the generator parameter $\theta$ to produce more realistic samples. In the analysis of theoretical properties of the original GAN, the authors proved that when the discriminator is optimal, minimising the GAN objective in Equation (2.2.1) is equivalent to minimising the Jensen-Shannon divergence.

GANs have been proved successful in training IGMs, breaking new ground in terms of visual fidelity and sampling speed. However, it suffers from a number of drawbacks such as mode collapsing and training instability. Moreover, the fact that the values of training loss cannot reliably reflect the quality of generated samples makes it difficult to judge whether the learning is completed. Having said that, GANs opened the door for training IGMs using an adversarial training algorithms with a learned comparison of real and fake samples.

Later studies realised that the discriminator can be replaced by a distance with superior theoretically properties for the comparison between two empirical distributions of real and fake samples. To formulate adversarial training algorithms, existing GAN frameworks typically seek a worst-case distance, i.e., a maximisation over a parametrised distance, to act as the discriminator, while the the role of the generator remains the same.

### 2.2.2 $f$-GAN

Nowozin et al. [118] proposed $f$-GAN whose training objective is derived from $f$-divergences. Given probability measures $\mu$ and $\nu$ associated with densities $p$ and $q$, both of which are absolutely continuous on $\mathcal{X}$, the $f$-divergence is defined as

$$D_f(\mu,\nu) = \int q f(\frac{p}{q}) d\mu, \tag{2.2.2}$$

where the function $f : \mathbb{R}_+ \to \mathbb{R}$ is convex, lower-semicontinuous function satisfying $f(1) = 0$, i.e., when $\mu = \nu$, any divergence $D_f(\mu,\nu) = 0$. The choice of $f$ function determines a specific divergence. For example, the Kullback-Leibler (KL) divergence corresponds to $f(u) = log(u)$.

According to the Fenchel's duality theorem (see [75]), every convex, lower-semicontinuous function has a *Fenchel conjugate* dual function $f^*$. In particular, the function $f$ can be alternatively expressed using its dual function $f^*$ as

$$f(u) = \sup_{g \in \mathcal{G}} \{gu - f^*(g)\}, \tag{2.2.3}$$

where $\mathcal{G}$ is the domain of $f^*$.

Plugging the above representation of $f$ in the definition of the $f$-divergence in Equation (2.2.2) to obtain a lower bound on the divergence (see [116] for details), we have

$$\begin{aligned} D_f(\mu,\nu) &= \int q \sup_{g \in \mathcal{G}} \{g\frac{p}{q} - f^*(g)\} d\mu && (2.2.4)\\ &\geq \sup_{g \in \mathcal{G}} \int \big[pg - qf^*(g)\big] d\mu && (2.2.5)\\ &= \sup_{g \in \mathcal{G}} \left\{ \int f d\mu - \int f^*(g) d\nu \right\}, && (2.2.6) \end{aligned}$$

where the lower bound is a result of the Jensen's inequality.

The $f$-divergence is reformulated into a maximisation over the difference between two expectations. Nowozin et al. [118] parametrised the function $f_\varphi : \mathcal{X} \to \mathbb{R}$ with parameter $\varphi$

in order to distinguish the fake samples from the real ones. We denote the model distribution of the fake samples as $v_\theta$, whose parameter $\theta$ is inherited from the generative model that generates the samples. Finally, we arrive at the objective function for $f$-GAN,

$$\inf_\theta \sup_\varphi \Big\{ \mathbb{E}_{x\sim\mu}[f_\varphi(x)] - \mathbb{E}_{y\sim v_\theta}[f_\varphi(y)] \Big\}. \tag{2.2.7}$$

At the first glance, it may seem counter-intuitive to maximise a lower bound on a divergence. It is worth to point out that the bound is maximised to match the divergences that compare how two distributions differ before being minimised over the generator parameters. With an appropriate function chosen for $f$, we can also recover the original GAN objective.

### 2.2.3 MMD-GAN

We start this section by formally defining the maximum mean discrepancy (MMD) which was initially used for two-sample test in statistics to distinguish two sets of finite samples. Let $x$ and $y$ be random variables defined on a topological space $\mathcal{X}$, with respective Borel probability measures $\mu$ and $\nu$. Given observations $X := \{x_i\}_{i=1}^M$ and $Y := \{y_i\}_{i=1}^N$, independently and identically distributed (i.i.d.) from $\mu$ and $\nu$. Let $\mathcal{F}$ be a class of functions $f : \mathcal{X} \to \mathbb{R}$. The MMD is defined as

$$\mathcal{M}_d(\mu,\nu) := \sup_{f\in\mathcal{F}} \Big\{ \mathbb{E}_x[f(x)] - \mathbb{E}_y[f(y)] \Big\}, \tag{2.2.8}$$

where $\mathbb{E}_x$ and $\mathbb{E}_y$ denote the expectations with respect to $\mu$ and $\nu$.

This is also known as the integral probability metric, see [115]. Gretton et al. [68] discussed the case when the class of functions is chosen to be in a Reproducing Kernel Hilbert Space (RKHS), in which any function $f$ can be alternatively expressed using a reproducing kernel function $k(x,x') : \mathcal{X}\times\mathcal{X} \mapsto \mathbb{R}$. Through kernel mean embedding, the squared MMD distance can be written using the kernel representation of $f$ as

$$\mathcal{M}_k(\mu,\nu) := \mathbb{E}_{x,x'}[k(x,x')] - 2\mathbb{E}_{x,y}[k(x,y)] + \mathbb{E}_y[k(y,y')], \tag{2.2.9}$$

where $x'$ is an independent sample of $x$ drawn from $\mu$, and $y'$ is an independent sample of $y$ from $\nu$. It is shown that $\mathcal{M}_k(\mu,\nu) = 0$ if and only if (iff) $\mu = \nu$, provided that $k$ is a characteristic kernel. For the detailed definitions and proofs, please refer to [68] section 2.2.

Gretton et al. [68] also obtained an unbiased empirical estimate of $\mathcal{M}_k$. For all samples in $x$ and $y$, we have

$$\mathcal{M}_u(\hat{\mu},\hat{\nu}) = \frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{j\neq i}^{m} k(x_i,x_j) - \frac{2}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} k(x_i,y_j) + \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i}^{n} k(y_i,y_j), \tag{2.2.10}$$

where $\hat{\mu}$ and $\hat{\nu}$ are the empirical distributions of $x$ and $y$.

Li et al. [104] proposed MMD-GAN based upon $\mathcal{M}_u$ in Equation (2.2.10). The discriminator in MMD-GAN intends to learn a worst-case distance by searching the space of all characteristic kernels. However, it is difficult to guarantee that the kernels learned by neural networks during training remain characteristic kernels. Li et al. [104] therefore utilised the result that, for an injective function $f$ and a characteristic kernel $k$, the composed kernel $\tilde{k} = k \circ f$ is still a characteristic kernel. Hence, the discriminator searches for an optimal characteristic kernel by learning an injective function.

The key challenge becomes how to ensure a learned function $f_\varphi$, parametrized by a neural network with parameters $\varphi$, to be injective. Recall that, for any injective function $f$, there exists a function $f^{-1}$ such that $f^{-1}(f(x)) = x$. This can be approximated by an encoder-decoder structure using two neural networks, i.e., $f_{\varphi_d}(f_{\varphi_e}(x)) = x$ where $\varphi := (\varphi_e, \varphi_d)$. Denoting the learned characteristic kernel as $\tilde{k}_{\varphi_e} := k(f_{\varphi_e}(x_i), f_{\varphi_e}(x_j))$, we finally arrive at the objective function of MMD-GAN,

$$\inf_{\theta}\sup_{\varphi} \mathcal{M}_u^{\tilde{k}_{\varphi_e}}(\mu,\nu_\theta) - \lambda\,\mathbb{E}_{y\sim\nu_\theta}||y - f_{\varphi_d}(f_{\varphi_e}(y))||^2, \tag{2.2.11}$$

where the second term ensures the injective property of $f_{\varphi_e}(x)$ to be held.

### 2.2.4 Optimal Transport

In recent years, Optimal Transport (OT) has become a promising distance to compare the difference between two distributions. In particular, it has been adopted as the objective function for adversarial training. In this section, we proceed to review the variants of GANs built on the theory of OT, for which we provide a detailed revision as it also lays the foundation for our own works in this thesis.

OT was first posed by French mathematician Gaspard Monge in 1781 [114]. Put simply, the concept of OT concerns with the problem of finding a way to move a pile of earth from a starting location to another at a target location with the least cost. This cheapest moving cost is regarded as the distance between the two piles of earth.

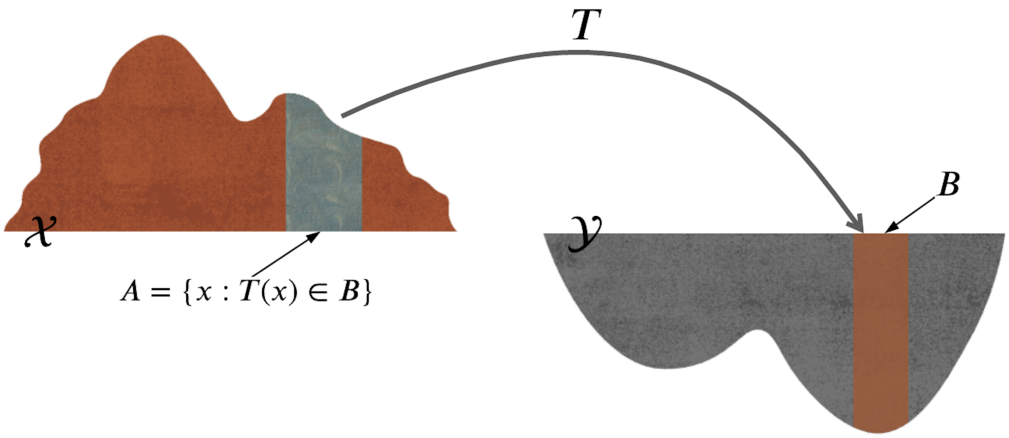



Fig. 2.4 Monge's transport map.

**Monge formulation**

We start with the definition of a Monge's transport map. Given probability measures $\mu$ on $\mathcal{X}$ and $\nu$ on $\mathcal{Y}$, we call $T : \mathcal{X} \to \mathcal{Y}$ a transport map if it transports mass from $x \in \mathcal{X}$ to $y \in \mathcal{Y}$ by

$$\nu(B) = \mu(T^{-1}(B)) \quad \text{for all } \nu\text{-measurable sets } B \subseteq \mathcal{Y}. \tag{2.2.12}$$

which is sometimes shorthanded as $\nu = T_{\#}\mu$ if above condition is satisfied.

An illustration of the Monge's transport map is shown in Figure 2.4. The mass moved away from $\mathcal{X}$ is always equivalent to the amount received on $\mathcal{Y}$, i.e., $\mu(A) = \nu(B)$ for any $\nu$-measurable set $B$ and $A = \{x : T(x) \in B\}$.

Let $\mu$, $\nu$, $x$, $y$, $\mathcal{X}$ and $\mathcal{Y}$ defined as above, and $c$ be a cost function $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, the **Monge's Optimal Transport** problem is formulated as

$$\mathcal{M}_c(\mu, \nu) := \inf_T \int_{\mathcal{X}} c(x, T(x)) d\mu(x), \tag{2.2.13}$$

where the cost of transporting one unit mass from $\mathcal{X}$ to $\mathcal{Y}$ is minimised over the $\mu$-measurable maps $T$ subject to $\nu = T_{\#}\mu$.

Note that the mass on $x$ is entirely mapped to $T(x)$. In another word, the Monge formulation does not permit any split of mass. This causes difficulties to guarantee the existence of transport maps that satisfy $\nu = T_{\#}\mu$. For example, there exists no transport plan $T$ that allows us to move the mass from $\mu = \frac{1}{2}\delta_{x_1} + \frac{1}{2}\delta_{x_2}$ for $x_1 \neq x_2$ to $\nu = \frac{1}{3}\delta_{y_1} + \frac{2}{3}\delta_{y_2}$, because $\nu(\{y_1\}) = \frac{1}{3}$ but no such value can be taken for $\mu(x)$ for any $x \in T^{-1}(y_1)$. However, if allowing splitting mass, we can transport one third of the mass on $x_1$ (i.e., $\frac{1}{6}$) to $y_1$ and the

rest to $y_2$. This relaxation to allow more flexible mass movements is the motivation of the Kantorovich's formulation.

### Kantorovich's formulation

To relax the constraint on the mass split in Monge's formulation, we consider a transport map $\pi \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$ that obeys the conservation law for the total mass transported. More specifically, $\pi$ satisfies the following constraints,

$$\pi(A \times \mathcal{Y}) = \mu(A), \pi(\mathcal{X} \times B) = \nu(B) \quad \text{for all measurable sets } A \subseteq \mathcal{X}, B \subseteq \mathcal{Y}. \tag{2.2.14}$$

Intuitively, this means that the total amount of mass removed from any measurable set $A$ has to equal to $\mu(A)$, and the total mount of mass received to any measurable set $B$ has to equal to $\nu(B)$.

Given probability measure $\mu$ on $\mathcal{X}$ and $\nu$ on $\mathcal{Y}$, a cost function $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, the **Kantorovich's Optimal Transport** Problem is defined as

$$\mathcal{W}_c(\mu, \nu) := \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} c(x,y) d\pi(x,y), \tag{2.2.15}$$

where $\Pi(\mu, \nu)$ denotes the set of *transport plans* between $\mu$ and $\nu$ that satisfy the constraints in Equation (2.2.14), and $d\pi(x,y)$ can be considered as the amount of mass transferred from $x$ to one or multiple locations of $y$.

It is well-understood that Kantorovich's and Monge's optimal transport problems do not always coincide. The advantage of Kantorovich's formulation is that when there exists a transport map for the Monge's problem, the optimal solution to the Kantorovich's problem is at least as good as that to the Monge's problem, see [160].

Furthermore, the Kantorovich's problem is convex and can be written as a linear programming problem. We devote the next section to explaining how this can be used to improve the computational efficiency of OT. In the remainder of the thesis, we use the term classic OT to refer to the Kantorovich's OT problem, and the primal form of OT to refer to Equation (2.2.15).

### Computation of Optimal Transport

Despite its intuitive formulation and appealing theoretical properties, the computation of OT involves solving an optimisation problem to which the solution can quickly become too expensive to compute when the size of the support or the data dimensions is large. To increase the applicability of primal form of OT in large-scale applications, Cuturi [37] introduced the

Sinkhorn distance to speed up the computation by adding an entropic regularisation to the problem.

As a classic linear programming problem, OT is always solved on a vertex of the feasible set of transport plans $\pi$. Such a vertex is a sparse matrix which is typically a solution difficult to reach. To facilitate the search, Cuturi [37] proposed to have a regularisation that encourages a smoother solution which is easier to obtain.

For two discrete marginal measures $\mu = \sum_i^N \delta_{x_i}$ on a finite set $\{x_i\}_{i=1}^N$ and $\nu = \sum_j^M \delta_{y_j}$ on a finite set $\{y_j\}_{j=1}^M$, we write the Kantorovich's OT problem as a linear programming problem:

$$\min_{\pi} \sum_{i=1}^{N}\sum_{j=1}^{M} c(x_i,y_j)\pi(x_i,y_j),$$

$$\text{subject to } \pi(x_i,y_j) \geq 0 \quad \text{for all } i = 1,...,N \text{ and } j = 1,...,M, \tag{2.2.16}$$

$$\sum_{i=1}^{N} \pi(x_i,y_j) = \nu, \text{ and } \sum_{j=1}^{M} \pi(x_i,y_j) = \mu, \tag{2.2.17}$$

where $\pi(x_i,y_j)$ is the joint distribution for $(x_i,y_j)$, and $c(x_i,y_j)$ is the cost for moving the mass on $x_i$ to $y_j$, both supported for all possible pairs $\{(x_i,y_j)\}_{i=1,j=1}^{N,M}$.

Denoting $\pi_{ij} = \pi(x_i,y_j)$, we define an entropic regularisation to the OT problem as the Shannon entropy of $\pi$: $H(\pi) := -\sum_{i,j} \pi_{ij} \log(\pi_{ij})$. For $\varepsilon > 0$, the regularized optimal transport problem then reads as

$$\mathcal{P}_{c,\varepsilon}(\mu,\nu) := \inf_{\pi \in \Pi(\mu,\nu)} \{\mathbb{E}^{\pi}[c(x,y)] - \varepsilon H(\pi)\}, \tag{2.2.18}$$

where $\mathbb{E}^{\pi}[f]$ denotes the expectation of an arbitrary function $f$ under transport plan $\pi$. The entropy $H(\pi)$ is be maximised when the table $\pi$ is smooth. Alternatively, we denote the set of regularised transport plans as $\Pi_{\varepsilon}(\mu,\nu)$, and rewrite the regularised OT as

$$\mathcal{W}_{c,\varepsilon}(\mu,\nu) := \inf_{\pi \in \Pi_{\varepsilon}(\mu,\nu)} \{\mathbb{E}^{\pi}[c(x,y)]\}. \tag{2.2.19}$$

Such an entropic regularisation permits the utilisation of an iterative algorithm, i.e., the Sinkhorn's fixed point iteration, to speed up the computation. Denoting $c_{ij} = c(x_i,y_j)$, we write the Lagrangian $\mathcal{L}(\pi,\alpha,\beta)$ of the regularised transport problem in Equation (2.2.19) with the constraints for the transport plans in Equation (2.2.16) as

$$\mathcal{L}(\pi,\alpha,\beta) = \sum_{ij} c_{ij}\pi_{ij} + \sum_{ij} \varepsilon \pi_{ij} \log(\pi_{ij}) + \alpha^{\top}(\pi \mathbf{1}_M - \mu) + \beta^{\top}(\pi^{\top}\mathbf{1}_N - \nu), \tag{2.2.20}$$

where $\mathbf{1}_M$ and $\mathbf{1}_N$ denote vectors filled with $M$ and $N$ ones, respectively.

Taking the partial derivative of $\mathcal{L}(\pi, \alpha, \beta)$ w.r.t $\pi_{ij}$ and set it to zero, we have

$$\begin{aligned}
0 &= c_{ij} + \varepsilon \log(\pi_{ij}) + \varepsilon + \alpha_i + \beta_j, \\
1 &= e^{c_{ij}} e^{\varepsilon \log(\pi_{ij})} e^{\varepsilon} e^{\alpha_i} e^{\beta_j}, \qquad \text{(take exponential on both sides)} \\
e^{-\varepsilon \log(\pi_{ij})} &= e^{c_{ij}} e^{\varepsilon} e^{\alpha_i} e^{\beta_j}, \\
\pi_{ij} &= e^{-\frac{1}{2} - \frac{\alpha_i}{\varepsilon}} e^{-\frac{c_{ij}}{\varepsilon}} e^{-\frac{1}{2} - \frac{\beta_j}{\varepsilon}}.
\end{aligned}$$

For $\varepsilon > 0$, $\pi_{ij}$ can be expressed as the so-called scaling form,

$$\pi_{ij} = a_i K_{i,j} b_j \quad \text{where} \quad \begin{aligned} K_{ij} &= e^{-\frac{c_{ij}}{\varepsilon}}, \\ a_i &= e^{-\frac{1}{2} - \frac{\alpha_i}{\varepsilon}}, \\ b_j &= e^{-\frac{1}{2} - \frac{\beta_j}{\varepsilon}}. \end{aligned}$$

Note that $K_{ij}$ is known because the cost function is pre-defined, whilst $a_i$ and $b_j$ are not due to the unknown Lagrangian coefficients $\alpha_i$ and $\beta_j$. Sinkhorn theorem [109] states that if all elements of matrix $K$ are strictly positive, then the problem can be solved with the Sinkhorn fixed point algorithm [147]. Initialising current iteration $l = 0$ and $b_j^l = 1$, the Sinkhorn algorithm finds the solution for the proposed problem by iterating the following computations until convergence:

$$a_i^{l+1} = \frac{1}{K_i b_j^l} \quad \text{and} \quad b_j^{l+1} = \frac{1}{K_j^\top a_i^{l+1}}. \tag{2.2.21}$$

The Sinkhorn algorithm has become a game changer for the applicability of the primal form of OT in modern machine learning. This attributes to the empirical observation that the number of iterations required for reaching an approximate solution of OT that leads to decent empirical results is relatively low, as well as the fact that both computations in Equation (2.2.21) only involve matrix-vector multiplication which can be computed efficiently on the modern GPUs.

### Duality of Optimal Transport

Every linear programming problem can be converted into a dual problem, providing a lower bound to the solution of the original minimisation problem. Given $\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y})$ and a cost function $c \in \mathcal{C}(\mathcal{X} \times \mathcal{Y})$ where $\mathcal{C}$ denotes the space of all continuous functions on $\mathbb{R}$, we

now write the dual problem of the primal form of the Kantorovich's OT problem:

$$\mathcal{W}_d := \sup\left\{ \int_{\mathcal{X}} \phi d\mu(x) + \int_{\mathcal{Y}} \psi d\nu(y) \middle| \phi \in \mathcal{C}_b(\mathcal{X}), \psi \in \mathcal{C}_b(\mathcal{Y}), \phi \oplus \psi \leq c \right\} \tag{2.2.22}$$

where $\mathcal{C}_b(\mathcal{X})$ denotes the space of all bounded continuous functions on $\mathbb{R}$ and $\phi \oplus \psi := \phi(x) + \psi(y)$. There is no duality gap if the primal value $\mathcal{W}_c$ of the OT problem equals the dual value $\mathcal{W}_d$, see analysis in [83, 134, 57, 49].

### 2.2.5 Wasserstein GAN

A special case for the classic OT defined in Equation (2.2.15) is when the spaces coincide, i.e., $\mathcal{X} = \mathcal{Y}$, and the cost $c$ thereby measures a distance between two samples in space $\mathcal{X}$. This leads to the definition of Wasserstein distance. Let $\mu$ and $\nu$ be two probability measures on $\mathcal{X}$, and $c$ be a cost function $c : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$. For $p \geq 1$, the $p^{th}$ order **Wasserstein distance** is given by

$$\mathcal{W}_p(\mu,\nu) := \left( \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{X}} c(x,y)^p d\pi(x,y) \right)^{\frac{1}{p}}, \tag{2.2.23}$$

where $\Pi(\mu,\nu)$ denotes the set of transport plans $\pi \in \mathcal{P}(\mathcal{X} \times \mathcal{X})$ with marginals $\mu$ and $\nu$.

When the cost function is defined as the L1 norm $c(x,y) = ||x-y||_1$, we have Wasserstein distance of order 1, namely the **Earth-Mover** or **Wasserstein-1 distance**:

$$\mathcal{W}_c^1(\mu,\nu) := \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{X}} ||x-y||_1 d\pi(x,y). \tag{2.2.24}$$

To formulate a min-max GAN objective function, Arjovsky et al. [8] proposed Wasserstein GAN (WGAN) utilising the special case of the duality theorem in Equation (2.2.22) for Wasserstein-1 distance, written as

$$\mathcal{W}_d^1(\mu,\nu) := \sup\left\{ \int_{\mathcal{X}} f(x) d\mu(x) - \int_{\mathcal{X}} f(x) d\nu(x) : ||f||_L \leq 1 \right\}, \tag{2.2.25}$$

where $||f||_L \leq 1$ represents the space of 1-Lipschitz functions $f : \mathcal{X} \mapsto \mathbb{R}$.

Given observations $x := \{x_i\}_{i=1}^M$ from $\mu$ and $y := \{y_i\}_{i=1}^N$ from $\nu$, the Wasserstein-1 distance can be interpreted as the difference between the expectations of two sets of independent samples. Arjovsky et al. [8] parametrised the Lipschitz functions $f_\varphi$ with parameter $\varphi$ to learn a worst-case distance by searching through the space of 1-Lipschitz functions. Denoting the distribution on samples generated from the generator $g_\theta$ as $\nu_\theta$ parametrised with $\theta$, we

write the WGAN objective function as

$$\inf_{\theta}\sup_{\varphi}\left\{\mathbb{E}_{x\sim\mu}[f_\varphi(x)]-\mathbb{E}_{y\sim\nu_\theta}[f_\varphi(y)]\right\}. \tag{2.2.26}$$

In WGAN, the 1-Lipschitz constraint is naively enforced by weight clipping on the discriminator parameter $\varphi$ to a fixed box of values $[-C,C]^N$ where $C$ is a constant. This is not ideal. Later work [70] proposed a softer version of the Lipschitz constraint by forcing the gradient norm to be in a unit ball, which further stabilised training and improved empirical results.

As one of the earliest GAN frameworks that explored alternative discriminators, the significance of WGAN lies in the demonstration that any worst-case distance between two distributions can be constructed as the discriminator in adversarial training, shedding light on the development of new adversarial training algorithms. Moreover, Wasserstein distance provides a more meaningful indicator on the perceptual quality than a binary classifier which is used as the discriminator in the original GAN. This is because a binary classifier is blind to the differences in generated images once their similarity to the ground truth exceeds a certain threshold, leading to a discriminator that's unable to tell the "good enough" samples apart.

### 2.2.6 Sinkhorn GAN

Instead of tackling the dual problem of OT, Genevay et al. [62] presented Sinkhorn GAN which takes advantage of speedy computation of the regularised OT in Equation (2.2.19). Due to the entropic term added to the classic OT, the property $\mathcal{W}_{c,\varepsilon}(\mu,\mu)=0$ does not hold in the regularised OT. To correct this bias, Genevay et al. [62] proposed the Sinkhorn divergence,

$$\widetilde{\mathcal{W}}_{c,\varepsilon}(\mu,\nu)=2\mathcal{W}_{c,\varepsilon}(\mu,\nu)-\mathcal{W}_{c,\varepsilon}(\mu,\mu)-\mathcal{W}_{c,\varepsilon}(\nu,\nu). \tag{2.2.27}$$

Furthermore, it is shown that $\widetilde{\mathcal{W}}_{c,\varepsilon}(\mu,\nu)\to 2\mathcal{W}_c(\mu,\nu)$ as $\varepsilon\to 0$, and $\widetilde{\mathcal{W}}_{c,\varepsilon}(\mu,\nu)\to \mathcal{M}_d(\mu,\nu)$ as $\varepsilon\to+\infty$ where $\mathcal{M}_d(\mu,\nu)$ indicates the MMD distance when kernel is chosen to be the same as the cost function in $\mathcal{W}_c(\mu,\nu)$, see Theorem 1 in [62].

The discriminator in Sinkhorn GAN learns a worst-case distance by parametrising the cost function in OT with parameter $\varphi$,

$$c_\varphi(x,y)=||f_\varphi(x)-f_\varphi(y)||, \tag{2.2.28}$$

where $f_\varphi:\mathcal{X}\mapsto\mathbb{R}^p$ and $p$ is an arbitrary dimensions of the output.

Sinkhorn GAN trains the generator and discriminator by optimising over the Sinkhorn divergence,

$$\inf_{\theta} \sup_{\varphi} \widetilde{\mathcal{W}}_{c_\varphi,\varepsilon}(\mu, \nu_\theta). \tag{2.2.29}$$

The OT distance in the Sinkhorn GAN is solved by the Sinkhorn algorithm described in a previous section. This proves the applicability of the Sinkhorn algorithm for solving the OT problem in practice. The formulation of Sinkhorn GAN is similar to that of MMD-GAN, both of which learn a worst-case distance by maximising the difference between the projections of two sets of samples.

In Chapter 3, we introduce the concept of Causal Optimal Transport for sequential learning, construct an adversarial algorithm for training dynamic IGMs using the dual form (**??**) of it, and investigate alternative formulations for bias correction in comparison to the Sinkhorn divergence (2.2.27). The resulting algorithm forms a foundation for the works presented in Chapter 4 and Chapter 5. In Chapter 6, however, we adopted the Sinkhorn GAN instead because the data concerned in the experiments is not sequential.

## 2.3 Evaluation

The evaluation of probabilistic generative models is known to be challenging. The complexity of the issue attributes to a number of factors. First, as a form of unsupervised learning, we do not have access to the corresponding true labels to compare against as we do in supervised learning. Second, generative models have a wide range of applications upon which the evaluation metrics are determined. Evaluation metrics for time series generation differ, as they should, from those for images. Third, different aspects of sample qualities must be considered, such as distortion of the reconstructions and diversity in a set of samples.

For PGMs in which likelihood functions are specified, it is a natural choice to directly compare the log-likelihood. However, log-likelihood has been shown to be poorly associated with human-perceived sample qualities. Models with low log-likelihood can produce great samples, whilst those with high log-likelihood may produce what are perceived as much worse by humans [167, 158]. In comparison, evaluating IGMs are less straightforward as the likelihood functions are usually intractable.

In the application of image and video generation, the gold standard for evaluation is whether the samples are sharp and realistic to human eyes. Whilst evaluation by human rating is sometimes provided in existing literature, the majority of works in this domain use numerical evaluations only. On one hand, due to the costly and slow process of collecting user ratings, it is not available to all members of the community. On the other hand, when the

number of participants is small, the results can change drastically [138] based on subjective judgements.

Commonly used numerical evaluation metrics in image and video generation roughly fall into two main categories: pixel-based and distribution-based comparison between the real and generated samples. The former includes classic metrics such as mean square error (MSE) computed over all pixels, and Peak signal-to-noise ratio (PSNR) [50] which calculates a log-scaled ratio of the highest value in all pixels over MSE. Although the pixel-based comparison serves well as a measure for distortion, it is known to be poorly correlated with human ratings for perceptual quality [200].

Recent advances in generative models have promoted new distribution-based metrics to be developed. The new metrics typically extract latent features from the real and generated samples using a pre-trained vision model, and compute a distance between two distributions, often assumed to be Gaussian, fitted using the two sets of latent features. An example of such metrics is Fréchet Inception Distance (FID) [73] which extracts the features using three network layers pre-trained on ImageNet and computes the Fréchet distance between the two distributions in the latent space. The Fréchet Video Distance (FVD) [164] is similar to FID, except that it is designed and tested for comparing video sequences by extracting latent features via pre-trained 3D convolutional networks. Their kernel counterparts KID and KVD [23] obtain the latent features in a similar manner, but compute the MMD (2.2.8) between two sets of features instead.

Some generative models suffer from mode collapse. This means that the trained models can only produce a subset of the training data, causing reduced variety in the samples. This phenomenon is particularly apparent with GANs [157, 150]. To measure the degree of mode collapse, Sajjadi et al. [137] proposed the precision and recall method which trains a classifier on the generated samples, and tests its accuracy on the real data and vice versa.

# Chapter 3

# COT-GAN: Generating Sequential Data via Causal Optimal Transport

## 3.1 Introduction

Dynamical data are ubiquitous in the world, including natural scenes such as video and audio data, and temporal recordings such as physiological and financial traces. Being able to synthesise realistic dynamical data is a challenging unsupervised learning problem and has wide scientific and practical applications. In recent years, training implicit generative models (IGMs) has proven to be a promising approach to data synthesis, driven by the work on generative adversarial networks (GANs) [67].

Nonetheless, training IGMs on dynamical data poses an interesting yet difficult challenge. On one hand, learning complex spatial structures of static images has already received significant effort within the research community. On the other hand, temporal dependencies are no less complicated since the dynamical features are strongly correlated with spatial features. Recent works, including [136, 190, 46, 172, 162], often tackle this problem by separating the model or loss into static and dynamic components.

In this chapter, we examine training dynamic IGMs for sequential data. We introduce a **new adversarial objective** that builds on optimal transport (OT) theory, and constrains the transport plans to respect *causality*: the probability mass moved to the target sequence at time $t$ can only depend on the source sequence up to time $t$, see [2, 12]. A reformulation of the causality constraint leads to a new adversarial training objective, in the spirit of [62] but tailored to sequential data. In addition, we demonstrate that optimising the original Sinkhorn divergence over mini-batches causes biased parameter estimation, and propose the **mixed Sinkhorn divergence** which mitigates this problem. Our new framework, Causal Optimal

Transport GAN (COT-GAN), outperforms existing methods on a wide range of datasets from traditional time series to video sequences.

## 3.2 Background

### 3.2.1 Causal Optimal Transport

We now focus on data that consists of $d$-dimensional (number of channels), $T$-step long sequences, so that $\mu$ and $\nu$ are distributions on the path space $\mathbb{R}^{d\times T}$. In this setting we introduce a special class of transport plans, between $\mathcal{X} = \mathbb{R}^{d\times T}$ and $\mathcal{Y} = \mathbb{R}^{d\times T}$, that will be used to define our objective function. On $\mathcal{X}\times\mathcal{Y}$, we denote by $x = (x_1, ..., x_T)$ and $y = (y_1, ..., y_T)$ the first and second half of the coordinates, and we let $\mathcal{F}^{\mathcal{X}} = (\mathcal{F}_t^{\mathcal{X}})_{t=1}^T$ and $\mathcal{F}^{\mathcal{Y}} = (\mathcal{F}_t^{\mathcal{Y}})_{t=1}^T$ be the canonical filtrations (for all $t$, $\mathcal{F}_t^{\mathcal{X}}$ is the smallest $\sigma$-algebra s.t. $(x_1, ..., x_T) \mapsto (x_1, ..., x_t)$ is measurable; analogously for $\mathcal{F}^{\mathcal{Y}}$).

A transport plan $\pi \in \Pi(\mu,\nu)$ is called causal if

$$\pi(dy_t|dx_1,\cdots,dx_T) = \pi(dy_t|dx_1,\cdots,dx_t) \qquad \text{for all } t = 1,\cdots,T-1. \tag{3.2.1}$$

The set of all such plans will be denoted by $\Pi^{\mathcal{K}}(\mu,\nu)$.

Roughly speaking, the amount of mass transported by $\pi$ to a subset of the target space $\mathcal{Y}$ belonging to $\mathcal{F}_t^{\mathcal{Y}}$ depends on the source space $\mathcal{X}$ only up to time $t$. Thus, a causal plan transports $\mu$ into $\nu$ in a non-anticipative way, which is a natural requirement in a sequential framework. In the present chapter, we will use causality in the sense of Equation (3.2.1). Note that, in the literature, the term causality is often used to indicate a mapping in which the output at a given time $t$ depends only on inputs up to time $t$.

Restricting the space of transport plans to $\Pi^{\mathcal{K}}$ in the OT problem Equation (2.2.15) gives the COT problem

$$\mathcal{K}_c(\mu,\nu) := \inf_{\pi\in\Pi^{\mathcal{K}}(\mu,\nu)} \mathbb{E}^{\pi}[c(x,y)]. \tag{3.2.2}$$

COT has already found wide application in dynamic problems in stochastic calculus and mathematical finance, see e.g. [3, 1, 2, 14, 10]. The causality constraint can be equivalently formulated in several ways, see [12, Proposition 2.3]. We recall here the formulation most suited for our purposes. Let $\mathcal{M}(\mathcal{F}^{\mathcal{X}},\mu)$ be the set of $(\mathcal{X},\mathcal{F}^{\mathcal{X}},\mu)$-martingales, and define

$$\begin{aligned}\mathcal{H}(\mu) := \{(h,M) :& h = (h_t)_{t=1}^{T-1},\ h_t \in \mathcal{C}_b(\mathbb{R}^{d\times t}),\ M = (M_t)_{t=1}^T \in \mathcal{M}(\mathcal{F}^{\mathcal{X}},\mu), \\ & M_t \in \mathcal{C}_b(\mathbb{R}^{d\times t}) \quad \text{for all } t = 1,...,T\},\end{aligned} \tag{3.2.3}$$

where, as usual, $\mathcal{C}_b(\mathbb{X})$ denotes the space of continuous, bounded functions on $\mathbb{X}$. Then, a transport plan $\pi \in \Pi(\mu,\nu)$ is causal if and only if

$$\mathbb{E}^{\pi}\left[\sum_{t=1}^{T-1} h_t(y_{\leq t})\Delta_{t+1}M(x_{\leq t+1})\right] = 0 \text{ for all } (h,M) \in \mathcal{H}(\mu), \tag{3.2.4}$$

where $x_{\leq t} := (x_1, x_2, \ldots, x_t)$ and similarly for $y_{\leq t}$, and $\Delta_{t+1}M(x_{\leq t+1}) := M_{t+1}(x_{\leq t+1}) - M_t(x_{\leq t})$. Therefore $\mathcal{H}(\mu)$ acts as a class of test functions for causality. Intuitively, causality can be thought of as conditional independence ("given $x_{\leq t}$, $y_t$ is independent of $x_{>t}$"), that can be expressed in terms of conditional expectations. This in turn naturally lends itself to a formulation involving martingales. Where no confusion can arise, with an abuse of notation we will simply write $h_t(y), M_t(x), \Delta_{t+1}M(x)$ rather than $h_t(y_{\leq t}), M_t(x_{\leq t}), \Delta_{t+1}M(x_{\leq t+1})$.

### 3.2.2 Regularised Causal Optimal Transport

In the same spirit of [62], we include an entropic regularisation in the COT problem (3.2.2) and consider

$$\mathcal{P}^{\mathcal{K}}_{c,\varepsilon}(\mu,\nu) := \inf_{\pi \in \Pi^{\mathcal{K}}(\mu,\nu)} \left\{\mathbb{E}^{\pi}[c(x,y)] - \varepsilon H(\pi)\right\}. \tag{3.2.5}$$

The solution to such problem is then unique due to strict concavity of $H$. We denote by $\pi^{\mathcal{K}}_{c,\varepsilon}(\mu,\nu)$ the optimiser to the above problem, and define the regularised COT distance by

$$\mathcal{K}_{c,\varepsilon}(\mu,\nu) := \mathbb{E}^{\pi^{\mathcal{K}}_{c,\varepsilon}(\mu,\nu)}[c(x,y)].$$

**Remark 3.2.1.** In analogy to the non-causal case, it can be shown that, for discrete measures $\mu$ and $\nu$ (as in practice), the following limits holds:

$$\mathcal{K}_c(\mu,\nu) \xleftarrow[\varepsilon\to 0]{} \mathcal{K}_{c,\varepsilon}(\mu,\nu) \xrightarrow[\varepsilon\to\infty]{} \mathbb{E}^{\mu\otimes\nu}[c(x,y)],$$

where $\mu \otimes \nu$ denotes the independent coupling.

We now prove the limits stated in Remark 3.2.1.

**Lemma 3.2.2.** Let $\mu$ and $\nu$ be discrete measures, say on path spaces $\mathcal{X}^T$ and $\mathcal{Y}^T$, with $|\mathcal{X}| = m$ and $|\mathcal{Y}| = n$ where $m$ and $n$ are positive integers. Then

$$\mathcal{K}_{c,\varepsilon}(\mu,\nu) \xrightarrow[\varepsilon\to 0]{} \mathcal{K}_c(\mu,\nu).$$

*Proof.* We mimic the proof of Theorem 4.5 in [2], and note that the entropy of any $\pi \in \Pi(\mu,\nu)$ is uniformly bounded:

$$0 \le H(\pi) \le C := m^T n^T e^{-1}. \tag{3.2.6}$$

This yields

$$\begin{aligned}\inf_{\pi\in\Pi^{\mathcal{K}}(\mu,\nu)} \mathbb{E}^{\pi}[c(x,y)] - \varepsilon C + \varepsilon H(\pi^{\mathcal{K}}_{c,\varepsilon}(\mu,\nu)) \le & \inf_{\pi\in\Pi^{\mathcal{K}}(\mu,\nu)} \{\mathbb{E}^{\pi}[c(x,y)] - \varepsilon H(\pi)\} \\ & + \varepsilon H(\pi^{\mathcal{K}}_{c,\varepsilon}(\mu,\nu)) \\ \le & \inf_{\pi\in\Pi^{\mathcal{K}}(\mu,\nu)} \mathbb{E}^{\pi}[c(x,y)] + \varepsilon H(\pi^{\mathcal{K}}_{c,\varepsilon}(\mu,\nu)).\end{aligned} \tag{3.2.7}$$

Now, note that $\inf_{\pi\in\Pi^{\mathcal{K}}(\mu,\nu)} \{\mathbb{E}^{\pi}[c] - \varepsilon H(\pi)\} = \mathcal{K}_{c,\varepsilon}(\mu,\nu) - \varepsilon H(\pi^{\mathcal{K}}_{c,\varepsilon}(\mu,\nu))$, and that, for $\varepsilon \to 0$, the LHS and RHS in Section 3.2.2 both tend to $\mathcal{K}_c(\mu,\nu)$. □

**Lemma 3.2.3.** Let $\mu$ and $\nu$ be discrete measures. Then

$$\mathcal{K}_{c,\varepsilon}(\mu,\nu) \xrightarrow[\varepsilon\to\infty]{} \mathbb{E}^{\mu\otimes\nu}[c(x,y)].$$

*Proof.* Being $\mu$ and $\nu$ discrete, $\mathbb{E}^{\pi}[c]$ is uniformly bounded for $\pi \in \Pi^{\mathcal{K}}(\mu,\nu)$. Therefore, for $\varepsilon$ big enough, the optimizer in $\mathcal{P}^{\mathcal{K}}_{c,\varepsilon}(\mu,\nu)$ is $\hat{\pi} := \arg\max_{\pi\in\Pi^{\mathcal{K}}(\mu,\nu)} H(\pi) = \mu\otimes\nu$, the independent coupling, for which $H(\mu\otimes\nu) = H(\mu) + H(\nu)$; see [159] and [66]. Therefore, for $\varepsilon$ large enough, we have $\mathcal{K}_{c,\varepsilon}(\mu,\nu) = \mathbb{E}^{\mu\otimes\nu}[c(x,y)]$. □

This means that the regularized COT distance is between the COT distance and the loss obtained by independent coupling, and is closer to the former for small $\varepsilon$. Optimizing over the space of causal plans $\Pi^{\mathcal{K}}(\mu,\nu)$ is not straightforward. Nonetheless, the following proposition shows that the problem can be reformulated as a maximization over non-causal problems with respect to a specific family of cost functions.

**Proposition 1.** The regularized COT problem (3.2.5) can be reformulated as

$$\mathcal{P}^{\mathcal{K}}_{c,\varepsilon}(\mu,\nu) = \sup_{l\in\mathcal{L}(\mu)} \mathcal{P}_{c+l,\varepsilon}(\mu,\nu), \tag{3.2.8}$$

where $\mathcal{P}_{c+l,\varepsilon}$ represents the regularised OT (2.2.18) with cost function $c : \mathcal{X}\times\mathcal{Y} \mapsto \mathbb{R}$ and

$$\mathcal{L}(\mu) := \left\{ \sum_{j=1}^{J}\sum_{t=1}^{T-1} h_t^j(y)\Delta_{t+1}M^j(x) : J\in\mathbb{N}, (h^j,M^j)\in\mathcal{H}(\mu) \right\}. \tag{3.2.9}$$

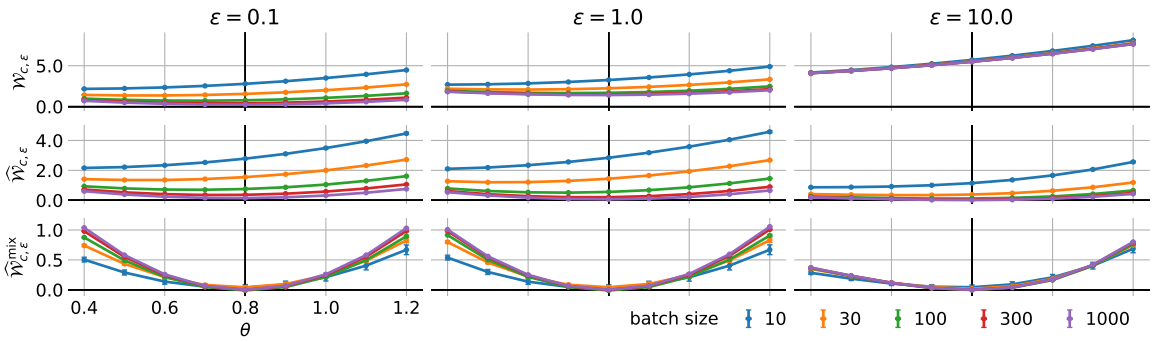



Fig. 3.1 Regularised distance (2.2.19), Sinkhorn divergence (2.2.27) and mixed Sinkhorn divergence (3.2.12) computed for mini-batches of size $m$ from $\mu$ and $\nu_\theta$, where $\mu = \nu_{0.8}$. Colour indicates batch size. Curve and errorbar show the mean and sem estimated from 300 draws of mini-batches.

This means that the optimal value of the regularized COT problem equals the maximum value over the family of regularized OT problems w.r.t. the set of cost functions $\{c+l : l \in \mathcal{L}(\mu)\}$. The proof for this result mimics the one that has been proven for the classic OT in [2].

Recall the alternative definition of regularised OT in Equation (2.2.19), Proposition 1 suggests the following worst-case distance between $\mu$ and $\nu$:

$$\sup_{l \in \mathcal{L}(\mu)} \mathcal{W}_{c+l,\varepsilon}(\mu,\nu), \tag{3.2.10}$$

as a regularized Sinkhorn distance that respects the causal constraint on the transport plans.

In the context of training a dynamic IGM, the training dataset is a collection of paths $\{x^i\}_{i=1}^N$ of equal length $T$, $x^i = (x_1^i,..,x_T^i)$, $x_t^i \in \mathbb{R}^d$. As $N$ is usually very large, we proceed as usual by approximating $\mathcal{W}_{c+l,\varepsilon}(\mu,\nu)$ with its empirical mini-batch counterpart. Precisely, for a given IGM $g_\theta$, we fix a batch size $m$ and sample $\{x^i\}_{i=1}^m$ from the dataset and $\{z^i\}_{i=1}^m$ from $\zeta$. Denote the generated samples by $y_\theta^i = g_\theta(z^i)$, and the empirical distributions by

$$\hat{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^m \delta_{x^i}, \quad \hat{\mathbf{y}}_\theta = \frac{1}{m}\sum_{i=1}^m \delta_{y_\theta^i}.$$

The empirical distance $\mathcal{W}_{c+l,\varepsilon}(\hat{\mathbf{x}},\hat{\mathbf{y}}_\theta)$ can be efficiently approximated by the Sinkhorn algorithm.

### 3.2.3 Reducing the bias with mixed Sinkhorn divergence

When implementing the Sinkhorn divergence in Equation (2.2.27) at the level of mini-batches, one canonical candidate clearly is

$$2\mathcal{W}_{c_\varphi,\varepsilon}(\hat{\mathbf{x}},\hat{\mathbf{y}}_\theta) - \mathcal{W}_{c_\varphi,\varepsilon}(\hat{\mathbf{x}},\hat{\mathbf{x}}) - \mathcal{W}_{c_\varphi,\varepsilon}(\hat{\mathbf{y}}_\theta,\hat{\mathbf{y}}_\theta), \tag{3.2.11}$$

which is indeed what is used in [62]. While the expression in (3.2.11) does converge in expectation to (2.2.27) for $m \to \infty$ ([60, Theorem 3]), it is not clear whether it is an adequate loss given data of fixed batch size $m$. In fact, we find that this is not the case, and demonstrate it here empirically.

**Example 3.2.4.** We build an example where the data distribution $\mu$ belongs to a parameterised family of distributions $\{\nu_\theta\}_\theta$, with $\mu = \nu_{0.8}$ (details in Section 3.4.3). As shown in Figure 3.1 (top two rows), neither the expected regularised distance (2.2.19) nor the Sinkhorn divergence (2.2.27) reaches minimum at $\theta = 0.8$, especially for small $m$. This means that optimizing $\nu$ over mini-batches will not lead to $\mu$.

Instead, we propose the following *mixed Sinkhorn divergence* at the level of mini-batches:

$$\widehat{\mathcal{W}}^{\text{mix}}_{c,\varepsilon}(\hat{\mathbf{x}},\hat{\mathbf{x}}',\hat{\mathbf{y}}_\theta,\hat{\mathbf{y}}'_\theta) := \mathcal{W}_{c,\varepsilon}(\hat{\mathbf{x}},\hat{\mathbf{y}}_\theta) + \mathcal{W}_{c,\varepsilon}(\hat{\mathbf{x}}',\hat{\mathbf{y}}'_\theta) - \mathcal{W}_{c,\varepsilon}(\hat{\mathbf{x}},\hat{\mathbf{x}}') - \mathcal{W}_{c,\varepsilon}(\hat{\mathbf{y}}_\theta,\hat{\mathbf{y}}'_\theta), \tag{3.2.12}$$

where $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ are the empirical distributions of mini-batches from the data distribution, and $\hat{\mathbf{y}}_\theta$ and $\hat{\mathbf{y}}'_\theta$ from the IGM distribution $\zeta \circ g_\theta^{-1}$. The idea is to take into account the bias within both the distribution $\mu$ as well as the distribution $\nu_\theta$ when sampling mini-batches.

Similar to (3.2.11), when the batch size $m \to \infty$, (3.2.12) also converges to (2.2.27) in expectation. So, the natural question arises: for a fixed $m \in \mathbb{N}$, which of the two does a better job in translating the idea of the Sinkhorn divergence at the level of mini-batches? Our experiments suggest that (3.2.12) is indeed the better choice. As shown in Figure 3.1 (bottom row), $\widehat{\mathcal{W}}^{\text{mix}}_{c,\varepsilon}$ finds the correct minimizer for all $m$ in Example 3.2.4. To support this finding, note that the triangular inequality implies

$$\mathbb{E}\left[\left|\mathcal{W}_{c_\varphi,\varepsilon}(\hat{\mathbf{x}},\hat{\mathbf{y}}_\theta) + \mathcal{W}_{c_\varphi,\varepsilon}(\hat{\mathbf{x}}',\hat{\mathbf{y}}'_\theta) - 2\mathcal{W}_{c,\varepsilon}(\mu,\nu)\right|\right] \le 2\mathbb{E}\left[\left|\mathcal{W}_{c_\varphi,\varepsilon}(\hat{\mathbf{x}},\hat{\mathbf{y}}_\theta) - \mathcal{W}_{c,\varepsilon}(\mu,\nu)\right|\right].$$

One can possibly argue that in (3.2.12) we are using two batches of size $m$, thus simply considering a larger mini-batch in (3.2.11), say of size $2m$, may perform just as well. However, we found this not to be the case and our experiments confirm that the mixed Sinkhorn divergence (3.2.12) does outperform (3.2.11) even when we allow for larger batch size. This reasoning can be extended by considering $\mathcal{W}_{c,\varepsilon}(.,.)$ with more terms for different

combinations of mini-batches. In fact, this is what is done in [141], which came to our attention after submitting this chapter for review. We have tested different variations in several experiments and while empirically there is no absolute winner, adding more mini-batches increases the computational cost; see Appendix 3.4.3.

### 3.2.4 COT-GAN: adversarial learning for sequential data

We now combine the results in Section 3.2.2 and Section 3.2.3 to formulate an adversarial training algorithm for IGMs. First, we approximate the set of functions (3.2.9) by truncating the sums at a fixed $J$, and we parameterize $\mathbf{h}_{\varphi_1} := (h^j_{\varphi_1})^J_{j=1}$ and $\mathbf{M}_{\varphi_2} := (M^j_{\varphi_2})^J_{j=1}$ as two separate neural networks, and let $\varphi := (\varphi_1, \varphi_2)$. To capture the adaptedness of those processes, we employ architectures where the output at time $t$ depends on the input only up to time $t$. The mixed Sinkhorn divergence between $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}_\theta$ is then calculated with respect to a parameterized cost function

$$c^{\mathcal{K}}_\varphi(x,y) := c(x,y) + \sum_{j=1}^{J}\sum_{t=1}^{T-1} h^j_{\varphi_1,t}(y)\Delta_{t+1}M^j_{\varphi_2}(x). \tag{3.2.13}$$

Second, it is not obvious how to directly impose the martingale condition (3.2.4), as constraints involving conditional expectations cannot be easily enforced in practice. Rather, we penalize processes $M$ for which increments at every time step are non-zero on average. For an $(\mathcal{X}, \mathcal{F}^{\mathcal{X}})$-adapted process $M^j_{\varphi_2}$ and a mini-batch $\{x^i\}^m_{i=1}$ ($\sim \hat{\mathbf{x}}$), we define the martingale penalization for $\mathbf{M}_{\varphi_2}$ as

$$p_{\mathbf{M}_{\varphi_2}}(\hat{\mathbf{x}}) := \frac{1}{mT}\sum_{j=1}^{J}\sum_{t=1}^{T-1}\left|\sum_{i=1}^{m}\frac{\Delta_{t+1}M^j_{\varphi_2}(x^i)}{\sqrt{\mathrm{Var}[M^j_{\varphi_2}]+\eta}}\right|,$$

where $\mathrm{Var}[M]$ is the empirical variance of $M$ over time and batch, and $\eta > 0$ is a small constant. Third, we use the mixed normalization introduced in (3.2.12). Each of the four terms is approximated by running the Sinkhorn algorithm on the cost $c^{\mathcal{K}}_\varphi$ for an a priori fixed number of iterations $L$.

Altogether, we arrive at the following adversarial objective function for COT-GAN:

$$\widehat{\mathcal{W}}^{\mathrm{mix},L}_{c^{\mathcal{K}}_\varphi,\varepsilon}(\hat{\mathbf{x}},\hat{\mathbf{x}}',\hat{\mathbf{y}}_\theta,\hat{\mathbf{y}}'_\theta) - \lambda p_{\mathbf{M}_{\varphi_2}}(\hat{\mathbf{x}}), \tag{3.2.14}$$

**Algorithm 1:** training COT-GAN by SGD

**Data:** $\{x^i\}_{i=1}^N$ (real data), $\zeta$ (probability distribution on latent space $\mathcal{Z}$)
**Parameters:** $\theta_0$, $\varphi_0$, $m$ (batch size), $\varepsilon$ (regularization parameter), $L$ (number of Sinkhorn iterations), $\alpha$ (learning rate), $\lambda$ (martingale penalty coefficient)
**Result:** $\theta$, $\varphi$
Initialize: $\theta \leftarrow \theta_0$, $\varphi \leftarrow \varphi_0$
**for** $k = 1, 2, \ldots$ **do**
  Sample $\{x^i\}_{i=1}^m$ and $\{x'^i\}_{i=1}^m$ from real data;
  Sample $\{z^i\}_{i=1}^m$ and $\{z'^i\}_{i=1}^m$ from $\zeta$;
  $(y_\theta^i, y_\theta'^i) \leftarrow (g_\theta(z^i), g_\theta(z'^i))$;
  Compute $\widehat{\mathcal{W}}^{\text{mix},L}_{c_\varphi^{\mathcal{K}},\varepsilon}(\hat{\mathbf{x}}, \hat{\mathbf{x}}', \hat{\mathbf{y}}_\theta, \hat{\mathbf{y}}'_\theta)$ (3.2.12) by the Sinkhorn algorithm, with $c_\varphi^{\mathcal{K}}$ given by (3.2.13);
  $\varphi \leftarrow \varphi + \alpha \nabla_\varphi \left( \widehat{\mathcal{W}}^{\text{mix},L}_{c_\varphi^{\mathcal{K}},\varepsilon}(\hat{\mathbf{x}}, \hat{\mathbf{x}}', \hat{\mathbf{y}}_\theta, \hat{\mathbf{y}}'_\theta) - \lambda p_{\mathbf{M}_{\varphi_2}}(\hat{\mathbf{x}}) \right)$;
  Sample $\{x^i\}_{i=1}^m$ and $\{x'^i\}_{i=1}^m$ from real data;
  Sample $\{z^i\}_{i=1}^m$ and $\{z'^i\}_{i=1}^m$ from $\zeta$;
  $(y_\theta^i, y_\theta'^i) \leftarrow (g_\theta(z^i), g_\theta(z'^i))$;
  Compute $\widehat{\mathcal{W}}^{\text{mix},L}_{c_\varphi^{\mathcal{K}},\varepsilon}(\hat{\mathbf{x}}, \hat{\mathbf{x}}', \hat{\mathbf{y}}_\theta, \hat{\mathbf{y}}'_\theta)$ (3.2.12) by the Sinkhorn algorithm, with $c_\varphi^{\mathcal{K}}$ given by (3.2.13); $\theta \leftarrow \theta - \alpha \nabla_\theta \left( \widehat{\mathcal{W}}^{\text{mix},L}_{c_\varphi^{\mathcal{K}},\varepsilon}(\hat{\mathbf{x}}, \hat{\mathbf{x}}', \hat{\mathbf{y}}_\theta, \hat{\mathbf{y}}'_\theta) \right)$;
**end**

where $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ are empirical measures corresponding to two samples of the dataset, $\hat{\mathbf{y}}_\theta$ and $\hat{\mathbf{y}}'_\theta$ are the ones corresponding to two samples from $\nu_\theta$, and $\lambda$ is a positive constant. We update $\theta$ to decrease this objective, and $\varphi$ to increase it.

While the generator $g_\theta : \mathcal{Z} \to \mathcal{X}$ acts as in classical GANs, the adversarial role here is played by $\mathbf{h}_{\varphi_1}$ and $\mathbf{M}_{\varphi_2}$. In this setting, the discriminator, parameterized by $\varphi$, learns a robust (worst-case) distance between the real data distribution $\mu$ and the generated distribution $\nu_\theta$, where the class of cost functions as in (3.2.13) originates from causality. The algorithm is summarized in Algorithm 3. Its time complexity scales as $\mathcal{O}((J+d)LTm^2)$ for each iteration.

## 3.3 Related work

Early video generation literature focuses on dynamic texture modeling [48, 154, 180]. Recent efforts in video generation within the GAN community have been devoted to designing GAN architectures of a generator and discriminator to tackle the spatio-temporal dependencies

separately, e.g., [172, 136, 162]. VGAN [172] explored a two-stream generator that combines a network for a static background and another one for moving foreground trained on the original GAN objective. TGAN [136] proposed a new structure capable of generating dynamic background as well as a weight clipping trick to regularize the discriminator. In addition to a unified generator, MoCoGAN [162] employed two discriminators to judge both the quality of frames locally and the evolution of motions globally.

The broader literature of sequential data generation attempts to capture the dependencies in time by simply deploying recurrent neural networks in the architecture [112, 52, 72, 190]. Among them, TimeGAN [190] demonstrated improvements in time series generation by adding a teacher-forcing component in the loss function. Alternatively, WaveGAN [46] adopted the causal structure of WaveNet [168]. Despite substantial progress made, existing sequential GANs are generally domain-specific. We therefore aim to offer a framework that considers (transport) causality in the objective function and is suitable for more general sequential settings.

Whilst our analysis is built upon [37] and [62], we remark two major differences between COT-GAN and the algorithm in [62]. First, we consider a different family of costs. While [62] learns the cost function $c(f_\varphi(x), f_\varphi(y))$ by parameterising $f$ with $\varphi$, the family of costs in COT-GAN is found by adding a causal component to $c(x,y)$ in terms of $\mathbf{h}_{\varphi_1}$ and $\mathbf{M}_{\varphi_2}$. The second difference is the mixed Sinkhorn divergence we propose, which reduces biases in parameter estimation and can be used as a generic divergence for training IGMs not limited to time series settings.

## 3.4 Experiments

### 3.4.1 Time series

We now validate COT-GAN empirically[1]. For times series that have a relatively small dimensionality $d$ but exhibit complex temporal structure, we compare COT-GAN with the following methods: **TimeGAN** [190] as reviewed in Section 3.3; **WaveGAN** [46] as reviewed in Section 3.3; and **SinkhornGAN**, similar to [62] with cost $c(f_\varphi(x), f_\varphi(y))$ where $\varphi$ is trained to increase the mixed Sinkhorn divergence with weight clipping. All methods use $c(x,y) = \|x-y\|_2^2$. The networks $h$ and $M$ in COT-GAN and $f$ in SinkhornGAN share the same architecture. Details of models and datasets are in Section A.2.1.

[1] Code and data are available at github.com/tianlinxu312/cot-gan

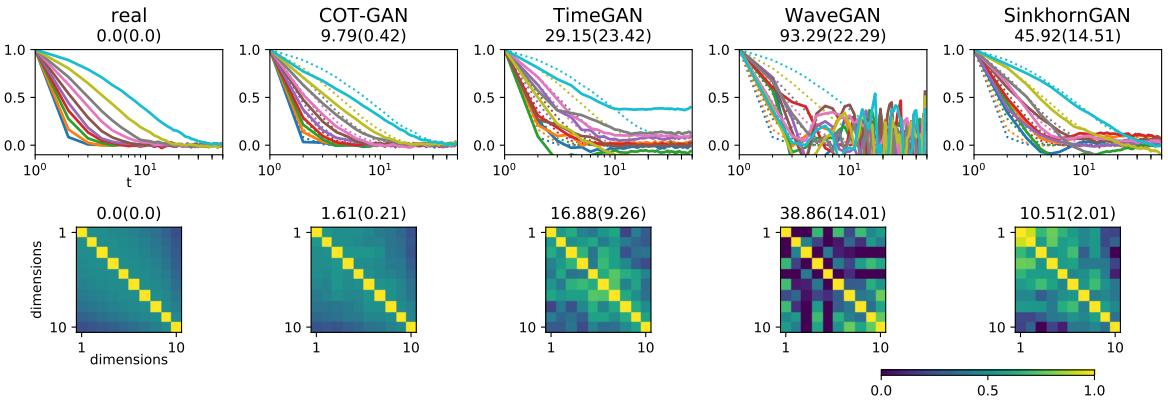



Fig. 3.2 Results on learning the multivariate AR(1) process. Top row shows the auto-correlation coefficient for each channel. Bottom row shows the correlation coefficient between channels averaged over time. The numbers on top of each panel are the mean and standard deviation (in brackets) of the sum of the absolute difference between the correlation coefficients computed from real (leftmost) and generated samples for 16 runs with different random seeds.

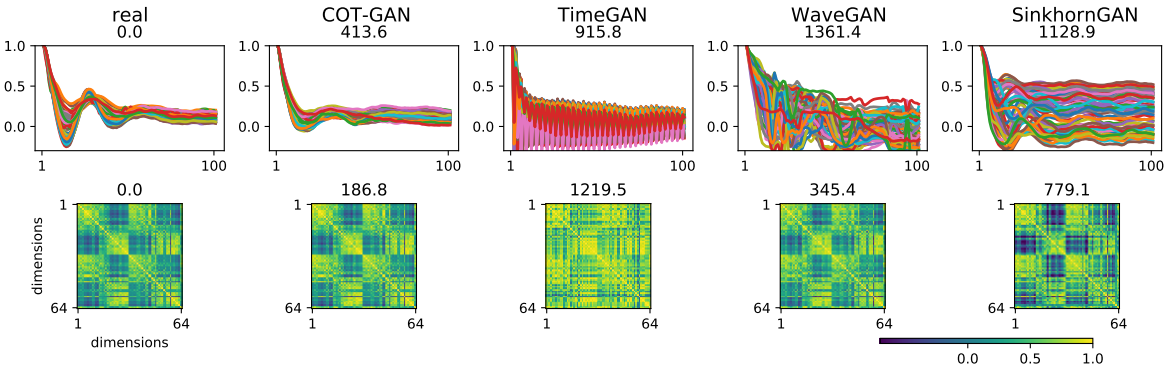



Fig. 3.3 Results on EEG data. The same correlations as Figure 3.2 are shown.

**Autoregressive processes.** We first test whether COT-GAN can learn temporal and spatial correlation in a multivariate first-order auto-regressive process AR(1).

For these experiments, we report two evaluation statistics: the sum of the absolute difference of the correlation coefficients between channels averaged over time, and the absolute difference between the correlation coefficients of real samples and those of generated samples. We evaluate the performance of each method by taking the mean and standard deviation of these two evaluation statistics over 16 runs with different random seeds.

In Figure 3.2, we show an example plot of results from a single run, as well as the evaluation statistics aggregated over all 16 runs on top of each panel. COT-GAN samples have correlation structures that best match the real data. Neither TimeGAN, WaveGAN nor SinkhornGAN captures the correlation structure for this dataset. The small standard

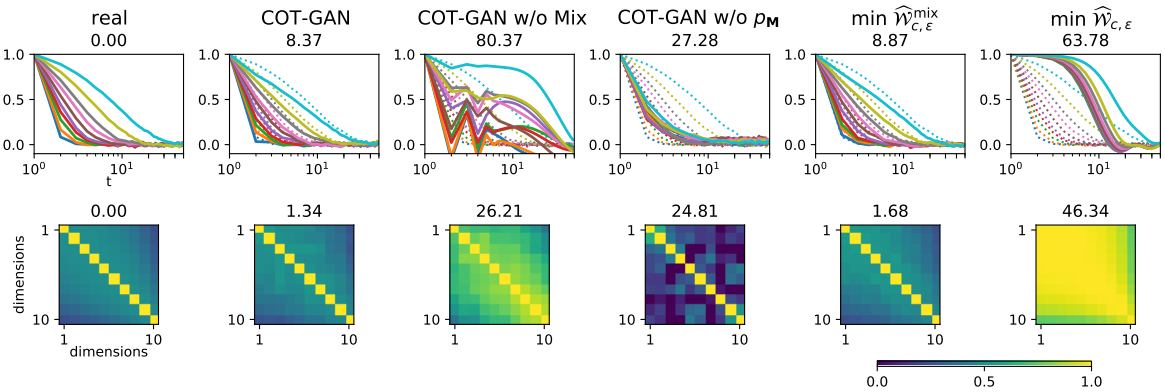



Fig. 3.4 Ablation investigation.

deviation of the evaluation statistics demonstrates that COT-GAN is the most stable model at least in the AR(1) experiment since it produces similar results from each run of the model.

**Noisy oscillations.** The noisy oscillation distribution is composed of sequences of 20-element arrays (1-D images) [182]. Figure A.1 in Section A.2.1 shows data as well as generated samples by different training methods. To evaluate performance, we estimate two attributes of the samples by Monte Carlo: the marginal distribution of pixel values, and the joint distribution of the location at adjacent time steps. COT-GAN samples match the real data best.

**Electroencephalography (EEG).** We obtain EEG dataset from [197] and take the recordings of all the 43 subjects in the control group with 80 trials under the matching condition (S2). For each subject, we choose 75% of the trials as training data and the remaining for evaluation, giving 2 841 training sequences and 969 test sequences in total. All data are subtracted by channel-wise mean, divided by three times the channel-wise standard deviation, and then passed through a tanh nonlinearity.

We truncated the sequences to only use the first 100 time steps, each of which has 64 channels. We compare performance of COT-GAN with respect to other baseline models. We evaluate model performance by investigating how well the generated samples match with the real data in terms of temporal and channel correlations in the same manner as done in the AR(1) experiment, except that the correlation matrix of the real samples is estimated using observations in this experiment since the true underlying data generating mechanism is unknown. See the results in Figure 3.3. We also examine how the coefficient $\lambda$ affects sample quality, see Section A.2.1. COT-GAN generates the best samples compared with other baselines across two metrics.

In addition, we provide an ablation investigation of COT-GAN, in which we study the impact of the components of the model by excluding each of them in the multivariate AR(1) experiment. In Figure 3.4, we compare the real samples with COT-GAN, COT-GAN using the original Sinkhorn divergence without the mixing, COT-GAN without the martingale penalty $p_{\mathbf{M}}$, direct minimization (without a discriminator) of the mixed and original Sinkhorn divergences from (3.2.12) and (3.2.11). We conclude that each component of COT-GAN plays a role in producing the best result in this experiment, and that the mixed Sinkhorn divergence is the most important factor for improvements in performance.

### 3.4.2 Videos

We train COT-GAN on animated Sprites [105, 131] and human action sequences [25]. We pre-process the Sprites sequences to have a sequence length of $T = 13$, and the human action sequences to have length $T = 16$. Each frame has dimension $64 \times 64 \times 3$. We employ the same architecture for the generator and discriminator to train both datasets. Both the generator and discriminator consist of a generic LSTM with 2-D convolutional layers. Details of the data pre-processing, GAN architectures, hyper-parameter settings, and training techniques are reported in Appendix A.2.2.

Baseline models chosen for the video datasets are **MoCoGAN** from [162], and direct minimization of the mixed Sinkhorn divergence (3.2.12), as it achieves a good result when compared to the other methods addressed in Figures 3.2 and 3.4. We show the real data and generated samples from COT-GAN side by side in Figure 3.5. Generated samples from all methods, without cherry-picking, are provided in Appendix A.3. The evaluation metrics we use to assess model performance are the Fréchet Inception Distance (FID) [73] which compares individual frames, the Fréchet Video Distance (FVD) [164] which compares the video sequences as a whole by mapping samples into features via pretrained 3D convolutional networks, and their kernel counterparts (KID, KVD) [23]. Previous studies suggest that FVD correlates better with human judgement than KVD for videos [164], whereas KID correlates better than FID on images [200].

In Table 5.1 the evaluation scores are estimated using 10,000 generated samples. For Sprites, COT-GAN performs better than the other two methods on FVD and KVD. However, minimization of the mixed Sinkhorn divergence produces slightly better FID and KID scores when compared to COT-GAN. The results in [164] suggest that FID better captures the frame-level quality, while FVD is better suited for the temporal coherence in videos. For the human action dataset, COT-GAN is the best performing method across all metrics except for KVD.

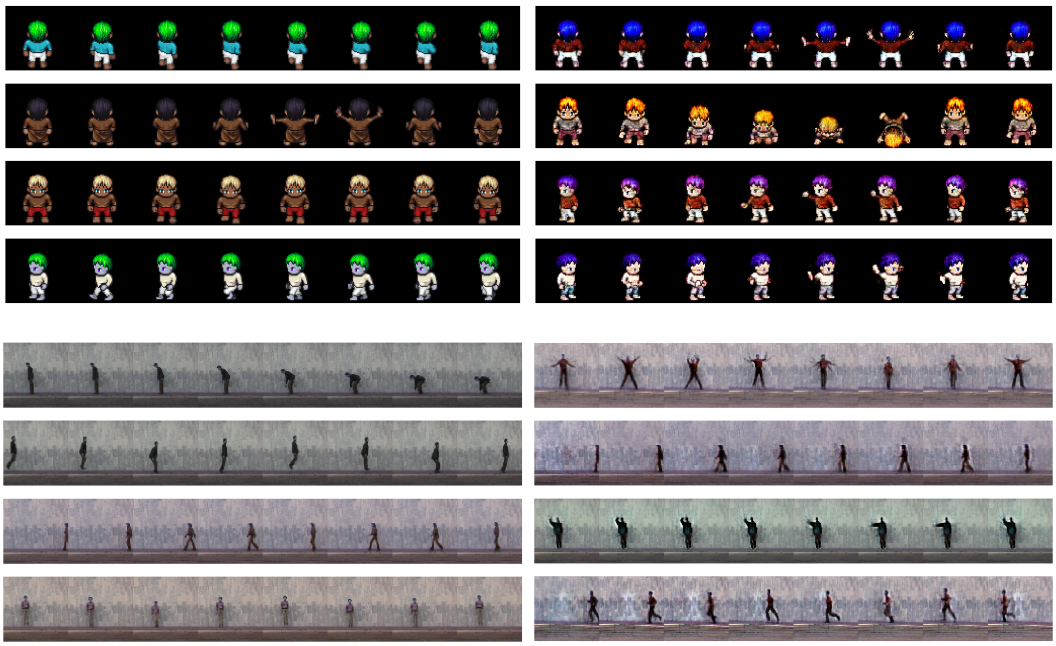


Fig. 3.5 Animated (top) and human (bottom) action videos. Left column reports real data samples, and right column samples from COT-GAN.

### 3.4.3 Mixed Sinkhorn divergence at various mini-batch levels

In the experiment mentioned in Example 3.2.4, we consider a set of distributions $\nu$'s as sinusoids with random phase, frequency and amplitude. We let $\mu$ be one element in this set whose amplitude is uniformly distributed between minimum 0.3 and maximum 0.8. On the other hand, for each $\nu$, the amplitude is uniformly distributed between the same minimum 0.3 and a maximum that lies in $\{0.4, 0.5, \ldots, 1.2\}$. Thus, the only parameter of the distribution being varied is the maximum amplitude. We may equivalently take the maximum amplitude as a single $\theta$ that parameterises $\nu_\theta$, so that $\mu = \nu_{0.8}$. Figure 3.1 illustrates that the sample Sinkhorn divergence Equation (3.2.11) (or regularised distance (2.2.19)) does not recover the optimiser 0.8, while the proposed mixed Sinkhorn divergence (3.2.12) does.

**Comparison of various implementations.** Motivated by Bellemare et al. [18], Salimans et al. [141] address the problem of bias in the mini-batch gradients of Wasserstein distance by proposing a mini-batch Sinkhorn divergence that is closely related to (3.2.12). We denote the implementation of a mini-batch Sinkhorn divergence in Salimans et al. [141] as

$$\begin{aligned}\widehat{\mathcal{W}}^6_{c,\varepsilon} := \mathcal{W}_{c,\varepsilon}(\hat{\mathbf{x}}, \hat{\mathbf{y}}_\theta) + \mathcal{W}_{c,\varepsilon}(\hat{\mathbf{x}}, \hat{\mathbf{y}}'_\theta) + \mathcal{W}_{c,\varepsilon}(\hat{\mathbf{x}}', \hat{\mathbf{y}}_\theta) + \mathcal{W}_{c,\varepsilon}(\hat{\mathbf{x}}', \hat{\mathbf{y}}'_\theta) \\ - 2\mathcal{W}_{c,\varepsilon}(\hat{\mathbf{x}}, \hat{\mathbf{x}}') - 2\mathcal{W}_{c,\varepsilon}(\hat{\mathbf{y}}_\theta, \hat{\mathbf{y}}'_\theta).\end{aligned}$$

Table 3.1 Evaluations for video datasets. Lower value indicates better sample quality.

| **Sprites** | FVD | FID | KVD | KID |
|---|---|---|---|---|
| MoCoGAN | 1 108.2 | 280.25 | 146.8 | 0.34 |
| $\min \widehat{\mathcal{W}}^{\text{mix}}_{c,\varepsilon}$ | 498.8 | **81.56** | 83.2 | **0.078** |
| COT-GAN | **458.0** | 84.6 | **66.1** | 0.081 |
| **Human actions** | | | | |
| MoCoGAN | 1 034.3 | 151.3 | 89.0 | 0.26 |
| $\min \widehat{\mathcal{W}}^{\text{mix}}_{c,\varepsilon}$ | 507.6 | 120.7 | **34.3** | 0.23 |
| COT-GAN | **462.8** | **58.9** | 43.7 | **0.13** |

In addition to (3.2.11) and (3.2.12), we further consider other possible variations of the Sinkhorn divergence at the level of mini-batches, including

$$\widehat{\mathcal{W}}^3_{c,\varepsilon} := 2\mathcal{W}_{c,\varepsilon}(\hat{\mathbf{x}}, \hat{\mathbf{y}}_\theta) - \mathcal{W}_{c,\varepsilon}(\hat{\mathbf{x}}, \hat{\mathbf{x}}') - \mathcal{W}_{c,\varepsilon}(\hat{\mathbf{y}}_\theta, \hat{\mathbf{y}}'_\theta)$$

and

$$\begin{aligned}\widehat{\mathcal{W}}^8_{c,\varepsilon} := {} & \mathcal{W}_{c,\varepsilon}(\hat{\mathbf{x}}, \hat{\mathbf{y}}_\theta) + \mathcal{W}_{c,\varepsilon}(\hat{\mathbf{x}}, \hat{\mathbf{y}}'_\theta) + \mathcal{W}_{c,\varepsilon}(\hat{\mathbf{x}}', \hat{\mathbf{y}}_\theta) + \mathcal{W}_{c,\varepsilon}(\hat{\mathbf{x}}', \hat{\mathbf{y}}'_\theta) \\ & - \mathcal{W}_{c,\varepsilon}(\hat{\mathbf{x}}, \hat{\mathbf{x}}') - \mathcal{W}_{c,\varepsilon}(\hat{\mathbf{y}}_\theta, \hat{\mathbf{y}}'_\theta) - \mathcal{W}_{c,\varepsilon}(\hat{\mathbf{x}}, \hat{\mathbf{x}}) - \mathcal{W}_{c,\varepsilon}(\hat{\mathbf{y}}_\theta, \hat{\mathbf{y}}_\theta).\end{aligned}$$

The superscripts in $\widehat{\mathcal{W}}^3_{c,\varepsilon}$, $\widehat{\mathcal{W}}^6_{c,\varepsilon}$ and $\widehat{\mathcal{W}}^8_{c,\varepsilon}$ indicate the number of terms used in the mini-batch implementation of the Sinkhorn divergence. In the same spirit, our choice of mixed Sinkhorn divergence $\widehat{\mathcal{W}}^{\text{mix}}_{c,\varepsilon}$ corresponds to $\widehat{\mathcal{W}}^4_{c,\varepsilon}$.

We compare the performance of all the variations in the low-dimensional applications of multivariate AR(1) and 1-D noisy oscillation (see Appendix A.2 for experiment details) in Figure 3.6 and Figure 3.7, and in the high-dimensional applications of Sprite animations and the Weizmann Action dataset in Table 3.2. The superscripts on COT-GAN correspond to the Sinkhorn divergence used in the experiments. We replace the COT-GAN objective (3.2.12) with (3.2.11) in the experiment of COT-GAN$^2$, with $\widehat{\mathcal{W}}^3_{c,\varepsilon}$ in COT-GAN$^3$, with $\widehat{\mathcal{W}}^6_{c,\varepsilon}$ in COT-GAN$^6$, and with $\widehat{\mathcal{W}}^8_{c,\varepsilon}$ in COT-GAN$^8$, respectively.

In the low-dimensional experiments, COT-GAN outperforms COT-GAN$^6$ on the 1-D noisy oscillation, but underperforms it on the multivariate AR(1) experiment. Both COT-GAN and COT-GAN$^6$ obtain better results than all other variations of the mini-batch Sinkhorn divergence. Given the low-dimensional results, we only compare COT-GAN and COT-GAN$^6$ in the high-dimensional experiments. As shown in Table 3.2, COT-GAN performs the best in all evaluation metrics except for KVD for Sprites animation. Both COT-GAN

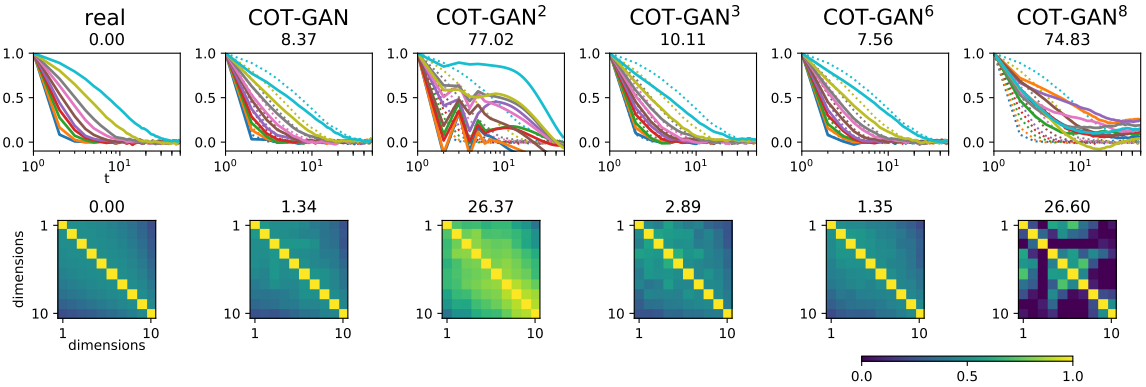



Fig. 3.6 Results on learning the multivariate AR(1) process.

and COT-GAN$^6$ perform better than MoCoGAN in these two tasks. However, because COT-GAN$^6$requires more mini-batches in the computation, it is about 1.5 times slower than COT-GAN.

Table 3.2 Evaluations for video datasets. Lower value indicates better sample quality.

| **Sprites** | FVD | FID | KVD | KID |
|---|---|---|---|---|
| MoCoGAN | 1 108.2 | 280.25 | 146.8 | 0.34 |
| COT-GAN$^6$ | 620.1 | 109.1 | **64.5** | 0.091 |
| COT-GAN | **458.0** | **84.6** | 66.1 | **0.081** |
| **Human actions** | | | | |
| MoCoGAN | 1 034.3 | 151.3 | 89.0 | 0.26 |
| COT-GAN$^6$ | 630.8 | 109.2 | 46.79 | 0.19 |
| COT-GAN | **462.8** | **58.9** | **43.7** | **0.13** |

## 3.5 Discussion

In this chapter, we introduce the use of causal transport theory in the machine learning literature. As already proved in other research fields, we believe it may have a wide range of applications here as well. The performance of COT-GAN already suggests that constraining the transport plans to be causal is a promising direction for generating sequential data. The approximations we introduce, such as the mixed Sinkhorn distance (3.2.12) and truncated sum in (3.2.9), are sufficient to produce good experimental results, and provide opportunities for more theoretical analyses in future studies. Directions of future development include ways to learn from data with flexible lengths, extensions to conditional COT-GAN, and improved

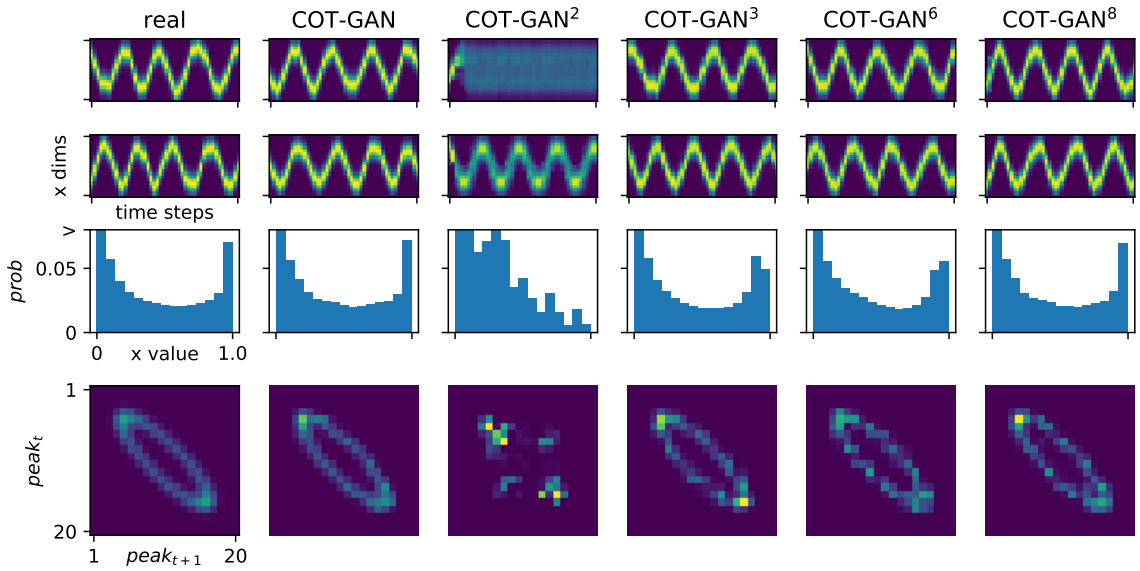



Fig. 3.7 1-D noisy oscillation. Top two rows show two samples from the data distribution and generators trained by different methods. Third row shows marginal distribution of pixels values (y-axis clipped at 0.07 for clarity). Bottom row shows joint distribution of the position of the oscillation at adjacent time steps.

methods to enforce the martingale property for the process $\mathbf{M}$ and better parameterize the causality constraint.

# Chapter 4

# Conditional COT-GAN with Kernel Smoothing

## 4.1 Introduction

Time series prediction is a challenging task. Given past observations, a desirable model should not only capture the distribution of features at each time step, but also predict its complex evolution over time. Autoregressive models which predict one time step after another seem to be a natural choice for learning such a task, see e.g. [41, 82, 119, 181]. However, the drawbacks of autoregressive models are the compounding error due to multi-step sampling and their high computational cost, see e.g. [82, 130]. Most existing models for time series prediction tend to ignore the temporal dependencies in sequences in the loss function, merely relying on certain specific network architectures, such as recurrent neural network (RNN) and 1D and 3D convolutional neural network (CNN), to capture the underlying dynamics, see e.g. [151, 4, 136, 172, 162]. For this learning task, the loss function used to compare prediction and real evolution plays a crucial role. However, a loss function that is blind to the sequential nature of data will almost certainly disappoint.

Yoon et al. [190] proposed TimeGAN to tackle this problem by introducing an auxiliary step-wise loss function to the original GAN objective, which indeed leads to more coherent and accurate predictions. More recently, the advances in the field of causal optimal transport (COT) have shown a promising direction for sequential modelling, see e.g. [13, 11, 126, 186]. This type of transport constrains the transport plans to respect temporal causality, in that the arrival sequence at any time $t$ depends on the starting sequence only up to time $t$. In this way, at every time we only use information available up to that time, which is a natural request in sequential learning. This is the foundation of COT-GAN [186], where the training objective

is tailored to sequential data. This proved to be an efficient tool, leading to generation of high-quality video sequences. Although the sharpness of single frames remains a challenge in video modelling, COT-GAN demonstrates that the evolution of motions can be reproduced in a smooth manner without further regularisation.

While COT-GAN is trained to produce sequences, the algorithm we propose here is learning *conditional sequences*, that is, how a sequence is likely to evolve given the observation of its past evolution. For this task, we employ a modification of the empirical measure that was introduced by Backhoff et al. [11] in the framework of *adapted Wasserstein ($\mathcal{AW}$) distance*. $\mathcal{AW}$-distance is the result of an optimal transport problem where the plans are constrained to be causal in both direction (so-called *bicausal optimal transport*); see [126, 127]. This turns out to be the appropriate distance to measure how much two processes differ, when we want to give importance to the evolution of information, see e.g. [14]. As noted in [127] and [11], the $\mathcal{AW}$-distance between a distribution and the empirical measure of a sample from it may not vanish while the size of the sample goes to infinity. To correct for this, Pflug and Pichler [127] proposed a convoluted empirical measure with a scaled smoothing kernel, while Backhoff et al. [11] suggested an adapted empirical measure obtained by quantization - both aiming to smooth the empirical measure in some way in order to yield a better convergence. In this chapter, we follow the approach of adapting the empirical measure by kernel smoothing as done in [127], and show that this smoothed empirical measure improves the performance of conditional COT-GAN.

The process described above gives rise to *kernel conditional COT-GAN*. The main contributions of the current chapter can then be summarised as follows:

- we extend the COT-GAN to a conditional framework, powered by an encoder-decoder style generator structure;
- we employ a new kernel empirical measure in the learning structure, which is a strongly consistent estimator with respect to COT;
- we show that our kernel conditional COT-GAN algorithm achieves state-of-the-art results for video prediction.

## 4.2 Framework

We are given a dataset consisting of $n$ i.i.d. $d$-dimensional sequences $\{x_1^i, \ldots, x_T^i\}_{i=1}^n$ where $T \in \mathbb{N}$ is the number of time steps and $d \in \mathbb{N}$ is the dimensionality at each time. This is thought of as a random sample from an underlying distribution $\mu$ on $\mathbb{R}^{d \times T}$, from which we want

to extract other sequences. More precisely, we want to learn the conditional distribution of $(x_{k+1},\dots,x_T)$ given $(x_1,\dots,x_k)$ under $\mu$, for any fixed $k \in \{1,\dots,T-1\}$. In the application of video prediction, an entire video contains $T$ frames, each of which has resolution $d$. The first $k$ frames of the video are taken as an input sequence, and later frames from time $k+1$ to $T$ are the target sequence. We will use the notation $x_{s:t} = (x_s,...,x_t)$, for $1 \le s \le t \le T$.

The conditional learning will be done via a conditional generative adversarial structure, based on a specific type of optimal transport tailored for distributions on path spaces, as introduced in the next section.

## 4.3 COT-GAN and CCOT-GAN

We now extend the analysis developed in Chapter 2 to a conditional framework for sequence prediction. Given the past history of a sequence up to time step $k$, the aim of CCOT-GAN is learning to predict the evolution from time step $k+1$ to $T$. The learning is done by stochastic gradient descent (SGD) on mini-batches. Given a sample $\{x^i_{1:T}\}_{i=1}^m$ from the dataset and a sample $\{z^i_{k+1:T}\}_{i=1}^m$ from a distribution $\zeta$ (noise) on some latent space $\mathcal{Z}$, we define the generator as a conditional model $g_\theta$, parameterised by $\theta$, which predicts the future evolution $\hat{x}^i_{k+1:T} = g_\theta(x^i_{1:k}, z^i_{k+1:T})$. The prediction $\hat{x}^i_{k+1:T}$ is then concatenated with the corresponding input sequence $x^i_{1:k}$ over the time dimension in order to be compared with the training sequence $x^i_{1:T}$ by the discriminator. We denote the empirical distributions of real and concatenated data by

$$\widehat{\mu} := \frac{1}{m}\sum_{i=1}^m \delta_{x^i_{1:T}}, \quad \widehat{\nu}^c_\theta := \frac{1}{m}\sum_{i=1}^m \delta_{\text{concat}(x^i_{1:k}, \hat{x}^i_{k+1:T})},$$

where $\widehat{\nu}^c_\theta$ incorporates the parameterisation of $g_\theta$ through $\{\hat{x}^i_{k+1:T}\}_{i=1}^m$, and concat(.) performs a concatenation operation over the channel dimension. Following COT-GAN's formulation of adversarial training, we arrive at the parameterised objective function for CCOT-GAN:

$$\widehat{\mathcal{W}}_{c^{\mathcal{K}}_\varphi,\varepsilon}(\widehat{\mu}, \widehat{\nu}^c_\theta) - \lambda\, p_{\mathbf{M}_{\varphi_2}}(\widehat{\mu}). \tag{4.3.1}$$

In the implementation of CCOT-GAN, the generator $g_\theta$ is broken down into two components: an encoder that learns the features of input sequences $\{x^i_{1:k}\}_{i=1}^m$ and a decoder that predicts future evolutions given the features of inputs and noise $\{z^i_{k+1:T}\}_{i=1}^m$. The discriminator role is played by $\mathbf{h}_{\varphi_1}$ and $\mathbf{M}_{\varphi_2}$, which are parameterised separately by two neural networks that respect temporal causality. These can be implemented as RNNs or 1D or 3D CNNs that are constrained to causal connections only, see Appendix B for details. We maximise the

objective function (4.3.1) over $\varphi$ to search for a robust (worst-case) distance between the two empirical measures $\widehat{\mu}$ and $\widehat{\nu}_\theta^c$, and minimise it over $\theta$ to learn a conditional model that produces sequential prediction.

## 4.4 Adapted empirical measure and KCCOT-GAN

It was noted by Backhoff et al. [11] and Pflug and Pichler [127] that the (classical) empirical measures are not necessarily consistent estimators with respect to distances originating from transport problems where transports plans respect causality constraints. The *nested distance* [126] or *adapted Wasserstein* distance [11] is the result of an optimal transport problem where plans are required to satisfy the causality constraint (3.2.1) as well as its symmetric counterpart, when inverting the role of $x$ and $y$:

$$\mathcal{AW}_c(\mu,\nu) := \inf\{\mathbb{E}^\pi[c(x,y)] : \pi \in \Pi^{\mathcal{K}}(\mu,\nu), \pi' \in \Pi^{\mathcal{K}}(\nu,\mu)\}, \tag{4.4.1}$$

where $\pi'(dx,dy) = \pi(dy,dx)$.

Now, for any measure $\mu$, and for the empirical measures $\widehat{\mu}_m$ relative to a random sample of size $m$ from $\mu$, it is known (see e.g. [54]) that

$$\mathcal{W}_c(\mu,\widehat{\mu}_m) \to 0 \quad \text{as } m \to \infty,$$

whereas [11, 127] observe that this is not necessarily true when substituting the Wasserstein distance $\mathcal{W}_c$ with the adapted Wasserstein distance $\mathcal{AW}_c$. This is of course undesirable, in particular thinking of the fact that the discriminator will evaluate discrepancies between real and generated measures by relying on empirical measures of the corresponding minibatches, see Section 4.3 and [186].

In [11] and [127], two different ways of adapting the empirical measure are suggested: by smoothing using a scaled kernel and by a quantization technique, respectively. The quantization technique [11] divides the data space into sub-cubes, and maps every value to the centre of the sub-cube to which it belongs. We did not adopt this approach for two reasons: first, the convergence property proved in Theorem 1.3 in [11] only holds when the number of sub-cubes is extremely small if the dimensionality of the data is large (typically a few hundreds). To see why too few sub-cubes can be problematic, consider this technique with two sub-cubes. This will map all data into only two possible values, which discards substantial information from the original data. Second, the quantization technique is non-differentiable, requiring an approximation so the gradients can flow back via

back-propagation in the stage of learning. We therefore adopt the kernel smoothing approach which we describe in detail in the remainder of this section.

For a probability measure $\mu$ with density $f$, and a density function $k_h(x) := \frac{1}{h}k(\frac{x}{h})$ where $h$ is the bandwidth parameter, the density estimator $\hat{f}$ is defined as

$$\hat{f}(x) = \int k_h(x-y)f(y)dy = f * k_h(x), \tag{4.4.2}$$

where $*$ denotes the convolution of densities.

Denoting the measure induced by density $k_h$ as $K^f$, we can write the convoluted measures with density $k_h$ as the weighted empirical measures of $\widehat{\mu}$ and $\widehat{\nu}^c_\theta$:

$$\widehat{\mu}^f := \widehat{\mu} * K^f = \sum_{i=1}^m w_i \delta_{x^i_{1:T}}, \tag{4.4.3}$$

$$\widehat{\nu}^{c,f}_\theta := \widehat{\nu}^c_\theta * K^f = \sum_{i=1}^m w_i \delta_{\mathrm{concat}(x^i_{1:k}, \hat{x}^i_{k+1:T})}, \tag{4.4.4}$$

where the weight $w_i$ is determined by $k_h$. Intuitively, this smooths the observations by taking a weighted average of all observations, typically with more influence from neighboring points.

Pflug and Pichler [127] proved that the adapted Wasserstein distance of the convoluted measures converges, i.e.,

$$P(\mathcal{AW}_c(\widehat{\mu}^f, \widehat{\nu}^{c,f}_\theta) > \varepsilon) \to 0 \quad \text{as } m \to \infty,$$

provided that

1. *the kernel $k_h$ is nonnegative and compactly supported on $\mathbb{R}^D$,*
2. *the density $f$ is bounded and uniformly continuous,*
3. *the bandwidth $h$ is a function of the sample size $m$ that satisfies*

$$h_m \to 0, \quad \frac{mh_m}{|\log h_m|} \to \infty, \quad \frac{|\log h_m|}{\log\log m} \to \infty, \\ \text{and} \quad mh_m \to \infty, \quad \text{as} \quad m \to \infty, \tag{4.4.5}$$

4. *the measures $\mu$ and $\nu$ are conditionally Lipschitz.*

For proofs and detailed discussions, please see Theorem 2 and 4 in [127].

Note that convergence result above is derived for the adapted Wasserstein distance $\mathcal{AW}_c$. In order to deduce the results on $\mathcal{W}_c^{\mathcal{K}}$, notice that

$$\mathcal{W}_c^{\mathcal{K}}(\mu,\nu) \leq \mathcal{AW}_c(\mu,\nu) \tag{4.4.6}$$

for any probability measures $\mu,\nu$ and any cost function $c$, given that the set of transports over which minimization is done for causal optimal transport is bigger than that for $\mathcal{AW}$-distance, cf. (3.2.2) and (4.4.1).

Relying on the above convergence result (4.4.6), we now introduce the CCOT-GAN with kernel smoothing (KCCOT-GAN). The objective function of KCCOT-GAN at the level of minibatches is computed on the adapted empirical measures:

$$\widehat{\mathcal{W}}_{c_\varphi^{\mathcal{K}},\varepsilon}(\widehat{\mu}^f,\widehat{\nu}_\theta^{c,f}) - \lambda p_{\mathbf{M}_{\varphi_2}}(\widehat{\mu}^f). \tag{4.4.7}$$

We maximise the objective function over $\varphi$ to search for a worst-case distance between the two adapted empirical measures, and minimise it over $\theta$ to learn a conditional distribution that is as close as possible to the real distribution. The algorithm is summarised in Algorithm 3. Its time complexity scales as $\mathcal{O}((J+2d)2LTm^2)$ in each iteration. The distance $\widehat{\mathcal{W}}_{c_\varphi^{\mathcal{K}},\varepsilon}(\widehat{\mu}^f,\widehat{\nu}_\theta^{c,f})$ is approximated by the means of the Sinkhorn algorithm iteratively with a fixed number of iterations, see Appendix A.

## 4.5 Implementation of KCCOT-GAN

The generator of KCCOT-GAN consists of an encoder that learns features from the input sequences, and a decoder that generates predictions conditioned on the input features and noise, supported by convolutional LSTM (convLSTM) [145]. The decoder was trained using a hierarchical version of the Teacher Forcing algorithm [184] which feeds the real values from observations as inputs during the training stage, in order to reduce the compounding error from multi-step predictions. To make it concrete, we proceed to formulate the implementation of KCCOT-GAN.

To avoid confusion, we refer to the entire input $x_{1:T}$ as the input sequence, and to the sequence $x_{1:k}$ upon which the prediction $x_{k+1:T}$ is made as the context sequence. Since the full input sequence is available to us at the stage of training, we first learn the hierarchical

**Algorithm 2:** training KCCOT-GAN by SGD

**Data:** $\{x^i_{1:T}\}_{i=1}^n$(data), $\zeta$(distribution on latent space)
**Parameters:** $\theta_0$, $\varphi_0$(initialisation of parameters), $m$(batch size), $\varepsilon$(regularisation parameter), $\alpha$(learning rate), $\lambda$(martingale penalty coefficient), $h$(bandwidth parameter)
**Result:** $\theta$, $\varphi$
Initialize: $\theta \leftarrow \theta_0$, $\varphi \leftarrow \varphi_0$
**for** $k = 1, 2, \ldots$ **do**
(1) Sample $\{x^i_{1:T}\}_{i=1}^m$ from real data; (2) Learn features from input sequences: $\{e^i_{1:T}\}_{i=1}^m \leftarrow f_{\theta_e}(\{x^i_{1:T}\}_{i=1}^m)$; (3) Sample $\{z^i_{k:T-1}\}_{i=1}^m$ from $\zeta$;
(4) Predict conditioned on features and inputs: $\{\hat{x}^i_{k+1:T}\}_{i=1}^m$ $\leftarrow f_{\theta_d}(\{e^i_{1:T}\}_{i=1}^m, \{x^i_{k:T-1}\}_{i=1}^m, \{z^i_{k:T-1}\}_{i=1}^m)$; (5) Obtain smoothed measures: $\widehat{\mu}^f$ and $\widehat{\nu}^{c,f}_\theta$; (6) Compute $\widehat{\mathcal{W}}_{c^{\mathcal{K}}_\varphi,\varepsilon}(\widehat{\mu}^f, \widehat{\nu}^{c,f}_\theta)$ by the Sinkhorn algorithm; (7) Update discriminator parameter:
$\varphi \leftarrow \varphi + \alpha\nabla_\varphi\Big(\widehat{\mathcal{W}}_{c^{\mathcal{K}}_\varphi,\varepsilon}(\widehat{\mu}^f, \widehat{\nu}^{c,f}_\theta) - \lambda p_{\mathbf{M}_{\varphi_2}}(\widehat{\mu}^f)\Big)$;
(8) Repeat step (2) - (6); (9) Update generator parameter:
$\theta \leftarrow \theta - \alpha\nabla_\theta\left(\widehat{\mathcal{W}}_{c^{\mathcal{K}}_\varphi,\varepsilon}(\widehat{\mu}^f, \widehat{\nu}^{c,f}_\theta)\right)$;
**end**

features of it through an encoder with $n$ layers,

$$\begin{aligned} e^1_{1:T} &= f_{\theta^1_e}(x_{1:T}), \\ e^2_{1:T} &= f_{\theta^2_e}(e^1_{1:T}), \\ &\vdots \\ e^n_{1:T} &= f_{\theta^n_e}(e^{n-1}_{1:T}). \end{aligned}$$

From here on, we denote the encoder as $f_{\theta_e}$ parametrized by $\theta_e := \{\theta^1_e, \theta^2_e, ..., \theta^n_e\}$, and the features extracted by the encoder as $e_{1:T} := \{e^1_{1:T}, ..., e^n_{1:T}\}$.

To deploy the teacher forcing algorithm, we make use of the hierarchical features as well as the input sequence. At time step $k+1$, we predict $\hat{x}_{k+1}$ conditioned on $(e_k, x_k)$, under the assumption that the feature $e_k$ contains all the information about the context sequence. Instead of feeding the prediction $\hat{x}_{k+1}$ back to the model to make next prediction, we continue to predict $\hat{x}_{k+2}$ conditioned on $(e_{k+1}, x_{k+1})$ in an effort to prevent the model to derail from the truth by making a mistake in an intermediate step. As a result, we train the model to predict $\hat{x}_{k+1:T}$ conditioned on $(e_{k:T-1}, x_{k:T-1})$. In the inference stage, however, we do not

have the information beyond the context sequence. The prediction is therefore completed in an auto-regressive manner.

Given Gaussian noise $z_{k:T-1}$, the decoder $f_{\theta_d}$ with $l$ layers for $l \geq n+1$ learns to predict the future steps by

$$\begin{aligned} d^1_{k+1:T} &= f_{\theta^1_d}(e^n_{k:T-1}, z_{k:T-1}), \\ &\vdots \\ d^{l-1}_{k+1:T} &= f_{\theta^{l-1}_d}(e^1_{k:T-1}, d^{l-2}_{k+1:T}) \\ \hat{x}_{k+1:T} &= f_{\theta^l_d}(x_{k:T-1}, d^{l-1}_{k+1:T}). \end{aligned}$$

As usual, the generator parameters $\theta := \{\theta_e, \theta_d\}$ and discriminator parameters $\varphi$ are learned on the level of mini-batches via SGD. To yield better convergence property, we smooth the mini-batches in each iteration using a scaled Gaussian kernel with zero mean,

$$k_h(x) = \frac{1}{h} e^{-\frac{x^2}{2h^2}}.$$

Differently from the technique of Gaussian blur widely used in image processing, see e.g. [71, 133, 117, 64], we apply a 3D scaled Gaussian kernel to both spatio and temporal dimensions. In another line of work, Zhang et al. [195] show that convoluting measures with a kernel density estimator is also a valid approach to tackle the problem of disjoint supports in divergence minimization.

The choices of the bandwidth parameter $h$ are restricted by the conditions in Equation (4.4.5). In the implementation, we relax this assumption by deploying a decaying bandwidth as a function of the number of the training iterations, rather than a function of sample size $m$. We realise that this simplification may lead to inferior theoretical guarantee of convergence. However, we will leave the exploration of a more appropriate approach to satisfy the theoretical assumptions to future research.

## 4.6 Related work

Video prediction is an active area of research. Methods relying on Variational inference [26] and VAE [91], e.g. SV2P [9], SVP-LP [41], VTA [88], and VRNN [30], have shown promising results. The majority of adversarial models adopted in this domain were trained on the original GAN objective [67] or the Wasserstein GAN objective [8], both of which

provide step-wise comparison of sequences. SAVP [100] combined the objective function of the original GAN and VAE to achieve the state of the art performance.

Substantial efforts have been devoted to designing specific architectures that tackle the spatio-temporal dependencies, e.g. [172, 136, 162, 35, 110, 171], and training schemes that facilitate learning, e.g. [110, 171, 4]. Whilst some works such as TGAN [136] and VGAN [172] combined a static content generator with a motion generator, others, e.g. [162, 35], designed two discriminators to evaluate the spatial and temporal components separately. Mathieu et al. [110] explored a loss that measures gradient difference at frame level on top of an adversarial loss trained with a multi-scale architecture. As a result, better performance was achieved in comparison to a simple mean square error loss commonly used in the literature. MCnet [171] extended [110] by adopting *convolutional long short-term memory (ConvLSTM)* [145] in the networks. Alternatively, 3D CNN with progressively growing training scheme [85] was also shown to be successful by FutureGAN [4].

However, it may not be sufficient to rely solely on the network architecture to capture the temporal structure of data. An important development in time series synthesis and prediction is the identification of more suitable loss functions. TimeGAN [190] combined the original GAN loss with a step-wise loss that computes the distance between the conditional distributions in a supervised manner. By matching a conditional model to the real conditional probability $p(x_t|x_{1:t-1})$ at every time step, it explicitly encouraged the model to consider the temporal dependencies in the sequence. In comparison, COT-GAN [186] explored a more natural formulation for sequential generation which leads to convincing results.

## 4.7 Experiments

We compare **KCCOT-GAN** to **CCOT-GAN** without kernel smoothing as an ablation study, to **SVP-LP** (Denton and Fergus [41]), to **SAVP** (Lee et al. [100]), and to **VRNN** (Castrejon et al. [30]), on three well-established video prediction datasets. The source code and video results are available at https://github.com/neuripss2020/kccotgan. In all our experiments, the choice of cost function is $c(x,y) = \sum_t \|x_t - y_t\|_2^2$, and initial bandwidth $h$ is 1.5 and is gradually decayed to 0.1 as training progresses. We select the first 15 frames and downsample them to a resolution of $64 \times 64$. We use the first 5 frames as the context sequence and the rest 10 frames as the target sequence. All results are evaluated on test sets. Note that the maximum number of hidden units used for the layers in the generator and discriminator networks is 256 for the **GQN Mazes** and **BAIR Push Small** datasets and 128 for the **Moving MNIST** dataset, due to the constraint of available computation power. This is at most half of the baseline model sizes. Although a compromised model capacity is expected, KCCOT-GAN

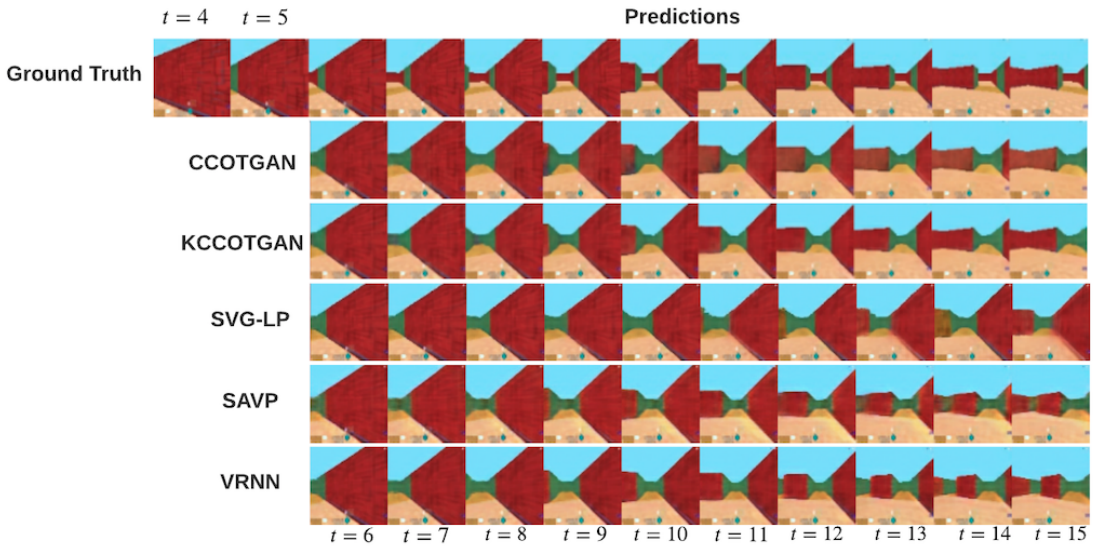



Fig. 4.1 GQN Mazes results on the test set. Only the last 2 frames from the context sequence are shown.

still produces excellent results on various tasks. Network architectures and more training details are given in Appendix B.

**GQN Mazes.** The GQN Mazes was first introduced by [51] for training agents to learn their surroundings by moving around. The dataset contains random mazes generated by a game engine. A camera traverses one or two rooms with multiple connecting corridors in each maze. The dataset comes with a training set that contains 900 sequences and a test set with a size of 120. The original sequences have a length of 300 and resolution of $84 \times 84$.

Figure 4.1 demonstrates that all models successfully captured the spatial structure in the frames well. However, predictions produced by SVG-LP lack of the evolution of motions, which is observed in many reproduced results of the model across various dataset. This could be attributed to the fact that SVG-LP is conditioned on a single frame from the previous time step, which makes it impossible for the model to pick up any information about past evolution. Visually, KCCOT-GAN and VRNN produced the sharpest frames out of all. Whilst samples from VRNN show more variations, those from KCCOT-GAN tend to be closer to the ground truth which may contribute to the better numerical evaluations in Table 5.1.

**BAIR Push Small.** Due to computation and storage constraint, we opted for this smaller version of the original BAIR Push dataset. The BAIR Push Small contains about 44,000 example with a resolution of $64 \times 64$. Each example shows a sequence of motions of robot arm pushing objects on a table.

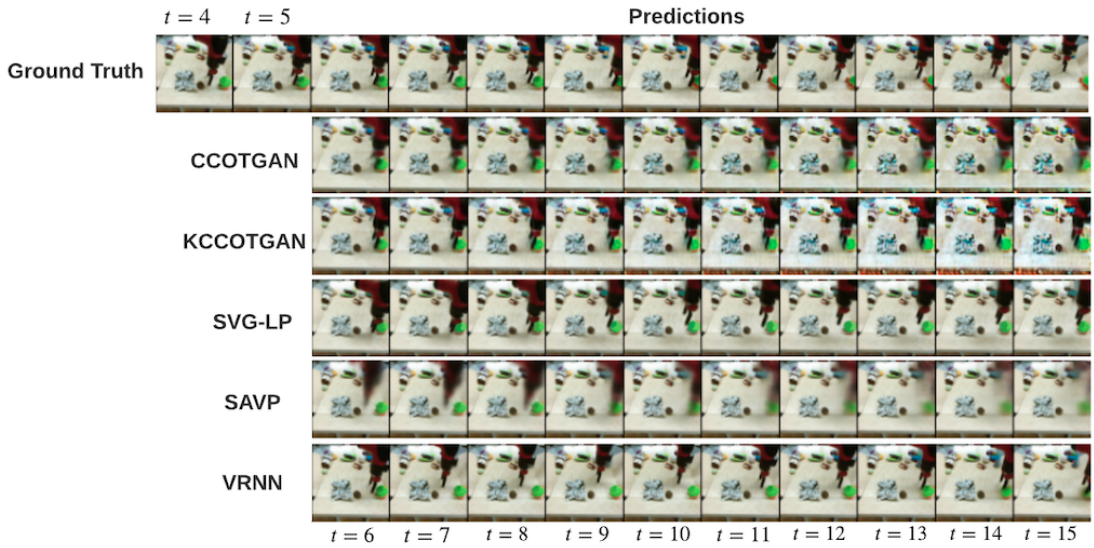



Fig. 4.2 BAIR Push Small results on the test set. Only the last 2 frames from the context sequence are shown.

For this dataset, the results from SVG-LP and VRNN are extremely good in terms of both the image quality and the variation in samples, see Figure 4.2. It is clearly a very difficult task to outperform these two baselines. On the other hand, SAVP has failed in producing high quality predictions.

On this dataset, although KCCOT-GAN underperforms the SVG-LP and VRNN baselines, we observe a clear improvement in sharpness from CCOT-GAN to KCCOT-GAN. As these two models share the same network structure and hyper-parameter settings, we can confirm that this improvement solely comes from the adaption of empirical measures via kernel smoothing.

**Moving MNIST Dataset.** Moving MINST [151] contains two digits that move with velocities sampled uniformly in the range of 2 to 6 pixels per frame and bounce within the edges of each frame. The dataset has 10000 sequences, of which we use 8000 for training and the rest for testing. Each of the original sequence contains 20 frames with resolution $64 \times 64$. Results are given in Table 5.1 and Appendix C.

**Evaluation.** We evaluate the video predictions using three metrics: Structural Similarity index [178] (SSIM, higher is better), Learned Perceptual Image Patch Similarity[196] (LPIPS, lower is better), Fréchet Video Distance [164] (FVD, lower is better).

The evaluation scores are reported in Table 5.1. We can see that KCCOT-GAN outperforms the baseline models on GQN Mazes dataset based on the three metrics. However,

VRNN are well ahead other models in BAIR Push Small dataset. The performances of VRNN and KCCOT-GAN on the Moving MMNIST dataset is reasonably close with KCCOT-GAN leading in SSIM and LPIPS but VRNN having better FVD score.

Table 4.1 Evaluations for video datasets. Lower values in the metrics indicate better sample quality for LPIPS and FVD, whereas higher values in SSIM are better.

| **GQN Mazes** | SSIM | LPIPS | FVD |
|---|---|---|---|
| SAVP | 0.49 | 0.077 | 488.35 |
| VRNN | 0.56 | 0.062 | 345.51 |
| SVG-LP | 0.43 | 0.094 | 575.22 |
| CCOT-GAN | 0.60 | 0.061 | 323.28 |
| KCCOT-GAN | **0.64** | **0.060** | **267.90** |
| **BAIR Push Small** | | | |
| SAVP | 0.502 | 0.090 | 280.32 |
| VRNN | **0.825** | **0.054** | **148.51** |
| SVG-LP | 0.822 | 0.059 | 158.80 |
| CCOT-GAN | 0.723 | 0.063 | 201.72 |
| KCCOT-GAN | 0.765 | 0.060 | 167.94 |
| **Moving MMNIST** | | | |
| SAVP | 0.571 | 0.123 | 129.33 |
| VRNN | 0.770 | 0.116 | **59.14** |
| SVG-LP | 0.668 | 0.160 | 101.39 |
| CCOT-GAN | 0.661 | 0.139 | 74.20 |
| KCCOT-GAN | **0.788** | **0.975** | 60.33 |

## 4.8 Discussion

In the present chapter we introduce KCCOT-GAN, the first algorithm for sequence prediction that is based on recently developed modifications of optimal transport specifically tailored for path spaces. For this we build on the results by [186], where COT was first applied for the task of sequential generation. Our experiments show the ability of KCCOT-GAN to not only capture the spatial structure in the frames, but also learn the complex dynamics evolving over time.

A limitation of the KCCOT-GAN algorithm is the lack of variations exhibited in the video sequences generated, in comparison to the baseline models that emphasise stochastic components in the model design. An improvement on KCCOT-GAN could be achieved by encoding more stochasiticity in the model. Another direction for future work is to explore

alternative choices of the kernel function convoluted over the empirical measures as well as a bandwidth parameter that better satisfies the conditions required for the convergence guarantee. One may also construct a learned kernel in a similar manner as done in MMD-GAN [104], whose parameters are updated along with those in the generator and discriminator.

# Chapter 5

# SPATE-GAN: Improved Generative Modelling of Dynamic Spatio-Temporal Patterns with an Autoregressive Embedding Loss

## 5.1 Introduction

Over the last decade, deep learning has emerged as a powerful paradigm for modelling complex data structures. It has also found successful applications in the video domain, for example for trajectory forecasting, video super-resolution or object tracking. Nevertheless, data observed over (discrete) space and time can take many more shapes than just RGB videos: many of the systems and processes governing our planet, from ocean streams to the spread of viruses, exhibit complex spatio-temporal dynamics. Current deep learning approaches often struggle to account for these, as a recent survey by Reichstein et al. [132] highlights. The authors call for more concerted research efforts aiming to improve the capacity of deep neural networks for modelling earth systems data. Recently, the emergence of physics-informed deep learning has reinforced the integration of physical constraints as a research domain [193, 129, 175, 86].

In this chapter, we propose a novel GAN tailored to the challenges of spatio-temporal complexities. We first devise a novel measure of spatio-temporal association—SPATE—expanding on the Moran's I measure of spatial autocorrelation (see definition in Section 5.3.1). SPATE uses the deviance of an observation from its space-time expectation, and compares it to neighbouring observations to identify regions of (relative) change and regions of (relative)

homogeneity over time. We propose three different approaches to calculate the space-time expectations, coming with varying assumptions and advantages for different applications. We then encode a SPATE-based embedding into COT-GAN [187] to formulate a new GAN framework, named SPATE-GAN. The motivation of choosing COT-GAN as the base model is that its principle of respecting temporal dependencies in sequential modelling is in line with our intuition for SPATE, see details in the methods section. Lastly, we test our approach on a range of different datasets. Specifically, we select data characterised by complex spatio-temporal patterns such as fluid dynamics [173, 175], disease spread [28] and global surface temperatures [148]. We observe that SPATE-GAN outperforms baseline models. This finding is particularly interesting as we do not change the architecture of the existing COT-GAN backbone, implying that our performance gains can be solely attributed to our novel SPATE-based embedding loss.

To summarise, the contributions of this study are as follows:

- We introduce SPATE, a new measure of spatio-temporal association, by expanding the intuition of the Moran's I metric into the temporal dimension.
- We introduce SPATE-GAN, a novel GAN for complex spatio-temporal data utilising SPATE to construct an embedding loss "nudging" the model to focus on the learning of autoregressive structures.
- We test SPATE-GAN against baseline GANs designed for image/video generation on datasets representing fluid dynamics, disease spread and global surface temperature. We show performance gains of SPATE-GAN over the baseline models.

## 5.2 Related work

### 5.2.1 Autocorrelation metrics for spatio-temporal phenomena

Analysing autoregressive patterns in spatial and spatio-temporal data has a long tradition in different academic domains (e.g, GIS, ecology) which over time developed diverse measures to describe these phenomena. The most commonly known of these metrics is the Moran's I index of global and local spatial autocorrelation. Originally proposed by Anselin [6], Moran's I identifies both homogeneous spatial clusters and outliers. Applications of the metric range from identifying rare earth contamination [191] to analysing land cover change patterns [38]. Throughout the years, Moran's I has also motivated several methodological expansions, analysing for example spatial heteroskedasticity [121] and local spatial dispersion [183]. Moran's I has also seen some expansions into the spatio-temporal domain. Matthews et al.

[111] use the metric iteratively to model disease spread over time. Lee and Li [101] and Gao et al. [58] propose novel spatio-temporal expansions of the Moran's I metric, returning static outputs at a purely spatial resolution. Siino et al. [146] design an extended Moran's I for spatio-temporal point processes. However, to the best of our knowledge, neither the Moran's I nor its spatio-temporal extensions have been applied to discrete spatio-temporal video data. It is evident that metrics of spatio-temporal autocorrelation can provide meaningful embeddings of complex data, capturing underlying patterns throughout a range of different application domains.

### 5.2.2 Deep learning & GANs for spatial and spatio-temporal data

Deep learning describes a powerful family of methods capable of dealing with the highly complex and non-linear nature of many real world spatial and spatio-temporal patterns [188, 7, 34, 16, 63]. Paradigms like physics-informed deep learning aim to devise methods which integrate (geo)physical constraints explicitly into neural network models [175]. There is also an increasing number of studies tackling specific challenges associated with geographic data: Mai et al. [108] and Yin et al. [189] propose context-aware vector embeddings for geographic coordinates. Zammit-Mangion et al. [192] propose to learn deep neural networks for spatial covariance patterns using injective warping functions. Intuitions for spatial autocorrelation, including the Moran's I metric, have been integrated into machine learning frameworks tackling model selection for ensemble learners [92], spatial representation learning [198], auxiliary task learning [93] or residual correlation graph neural networks [81]. All these studies highlight the benefits of explicitly encoding spatial context into neural networks to improve performance.

Narrowing down on the GAN context specifically, we find that spatio-temporal applications have mostly focused on video data [187, 163, 87]. Beyond this, GANs have been used for conditional density estimation of traffic [199], trajectory prediction [79] or extreme weather event simulation [94]. Nevertheless, to the best of our knowledge, metrics capturing spatio-temporal autocorrelation have never been integrated into GANs. As previous studies highlight the value on encoding spatial context, this work seeks to provide a first-principle approach of integrating metrics of spatio-temporal autocorrelation into GANs for modelling of complex spatio-temporal patterns.

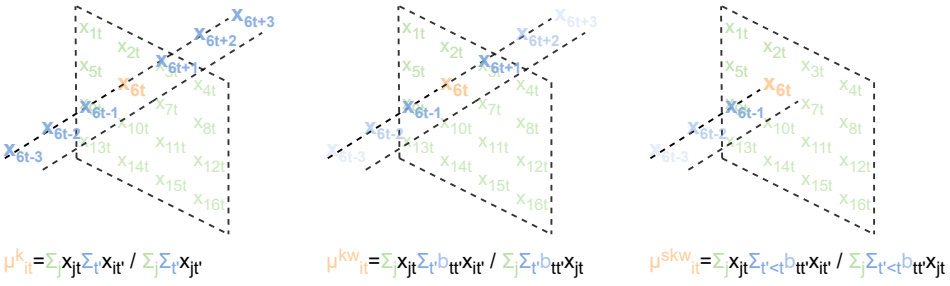
### 5.2.3 Embedding loss functions

SPATE-GAN integrates spatio-temporal metrics into COT-GAN as embeddings upon which the loss function is computed. We refer such loss functions as embedding losses which

have become popular in computer vision over the last years: Ghafoorian et al. [65] use embedding losses to improve GAN-based lane detection. Filntisis et al. [53] use visual-semantic embedding losses to improve predictions of bodily expressed emotions. Wang et al. [174] introduce CLIFFNet, utilising hierarchical embeddings of depth maps for molecular depth estimation. Bailer et al. [15] introduce a threshold loss to improve optical flow estimation. It is clear that embedding losses have shown great potential for particularly challenging visual problems, especially those involving complex spatio-temporal dynamics. They have led to desirable outcomes, such as improving training stability or facilitating the recognition of specific patterns in the data.

## 5.3 Methods

### 5.3.1 SPATE: Spatio-temporal association




**(a) Kulldorff spatio-temporal expectation** **(b) Kulldorff-weighted spatio-temporal expectation** **(c) Sequential Kulldorff-weighted spatio-temporal expectation**

Fig. 5.1 Illustrating the three proposed options to obtain *spatio-temporal expectations* $\mu_{it}$ used in the computation of SPATE for single-channel data.

For a static, discrete spatial pattern (e.g, a grid of pixels forming an image) consisting of continuous values $x \in \mathbb{R}^n$ where $n \in \mathbb{N}$, we use $x_i$ for $i \in \{1,\ldots,n\}$ to represent the $i$'th pixel value on a regular grid. The local Moran's I statistic $I_i$ is computed as follows:

$$I_i = (n_i - 1)\frac{z_i}{\sum_{j=1}^{n_i} z_j^2} \sum_{j=1, j\neq i}^{n_i} w_{i,j} z_j \tag{5.3.1}$$

where $z_i = x_i - \bar{x}$ is the deviance of observation $x_i$ from the global mean $\bar{x}$, $n_i$ is the number of spatial neighbours of pixel $x_i$, $j$ indexes neighbours of $x_i$ for $j \in \{1,\ldots,n_i\}$ and $j \neq i$, and $w_{i,j}$ is a binary spatial weight matrix, indicating spatial neighbourhood of observations $i$ and $j$. $I_i$ can be interpreted as a measure of similarity to neighbouring pixels: positive values imply homogeneous clusters, while negative values suggest outliers, change patterns or edges.

Now, let us assume that we observe a sequence of spatial patterns over time $t$: $x = (x_1, ..., x_T) \in \mathbb{R}^{n \times T}$ where $n$ is the dimensionality of $x_t$ at each time $t$ and $T$ is the length of the sequence. Of course, a naive adoption of the approach above is simply to ignore the time component of a sequence and compute the local Moran's I values $I_{it}$ around pixel $i$ using mean values $\bar{x}_t$ at each time $t$. Unfortunately, this approach would strictly separate spatial and temporal effects. In fact, a much more realistic assumption is that space and time are not separable, but do in fact interact and form joint patterns. For this reason, we expand the concept of Moran's I for spatio-temporal expectations. First, we follow the intuition outlined in Kulldorff et al. [97] and define expected values of $\mu_{it}(x)$. We refer to this approach as Kulldorff spatio-temporal expectation ("$k$"):

$$\mu_{it}^{(k)} = \frac{\sum_j x_{jt} \sum_{t'} x_{it'}}{\sum_j \sum_{t'} x_{jt'}}. \tag{5.3.2}$$

$\mu_{it}^{(k)}$ in (5.3.2) involves utilising all spatial units (pixels) available at time step $t$ and across all time steps at a single spatial unit (pixel position) $i$. This computation of the *spatio-temporal expectations* assumes independence of space and time, and thus the residual $z_{it} = x_{it} - \mu_{it}^{(k)}$ can be thought of as a local measure of space-time interaction at pixel $i$ and time $t$. Moreover, this formulation of $\mu_{it}$ makes two critical assumptions: (1) Different time steps are equally important, irrespective of how distant they are from the current time step. (2) At each time step, we assume availability of the whole time series (i.e. looking into the future is possible). We can modify the computation of $\mu_{it}$ by imposing alternative assumptions.

First, assuming that distant time steps have less significant impact on the current time step, we can integrate temporal weights into the computation, and apply decreasingly lower weights to more distant time steps. For example, one can consider an exponential kernel:

$$\mu_{it}^{(kw)} = \frac{\sum_j x_{jt} \sum_{t'} b_{tt'} x_{it'}}{\sum_j \sum_{t'} b_{tt'} x_{jt'}}, \tag{5.3.3}$$

where $b_{tt'} = \exp(-|t - t'|/l)$ and $l$ is the lengthscale of the exponential kernel. We refer to this approach as Kulldorff-weighted spatio-temporal expectation ("$kw$").

Second, we can restrict the computation of $\mu_{it}$ at time step $t$ to only account for time steps $< t$, so that the expectation at each time step is independent of future observations. Thus, the computation respects the generating logic of sequential data. We refer to this last approach as Kulldorff-sequential-weighted spatio-temporal expectation ("$ksw$"):

$$\mu_{it}^{(ksw)} = \frac{\sum_j x_{jt} \sum_{t'<t} b_{tt'} x_{it'}}{\sum_j \sum_{t'<t} b_{tt'} x_{jt'}}. \tag{5.3.4}$$

Note that for the first time step $t = 0$, we cannot access past time steps to calculate spatio-temporal expectations $\mu_{i0}^{(ksw)}$.

We can now simply plug in our new spatio-temporal expectations into the Moran's I metric at time $t$ by replacing spatial only expectations with a spatio-temporal expectation of our choice. As such, we define our novel measure of **spa**tio-**te**mporal association (SPATE),

$$S_{it}(x,w) = (n_i - 1)\frac{z_{it}}{\sum_{j=1}^{n_x} z_{jt}^2} \sum_{j=1, j\neq i}^{n_x} w_{i,j} z_{jt} \tag{5.3.5}$$

where $z_{it} = x_{it} - \mu_{it}$ and $\mu_{it}$ can be any option from $\mu_{it}^{(k)}$, $\mu_{it}^{(kw)}$ and $\mu_{it}^{(ksw)}$. When using $\mu_{it}^{(ksw)}$, SPATE does not return values for $t = 0$, as no previous time steps are available to calculate spatio-temporal expectations. See Fig 5.1 for an illustration of the three proposed options.

SPATE measures spatio-temporal autocorrelation at the input resolution. Its behaviour can be closely related to that of the Moran's I metric. While Moran's I evaluates the deviance $z_i$ between each pixel and the spatial expectation, SPATE does so by using the spatio-temporal expectation $z_{it}$. Like Moran's I, SPATE acts as a detector of spatio-temporal clusters and change patterns. Like Moran's I, SPATE identifies positive and negative space-time autocorrelation, i.e. homogeneous areas of similar behaviour and outliers that behave differently from their immediate neighbourhood. The difference between Moran's I and SPATE is that the later explicitly captures space-time interactions. For example, if pixel $x_i$ and all other data points (not just its neighbours) are increased at a given time step $t$, Moran's I at time $t$ (for all points) will be high, but SPATE will not be. SPATE of $x_{it}$ will be high if *(a)* pixel $x_{it}$ is high compared to its expectation at the same time $t$, *(b)* its neighbours are high compared to their expectations, while *(c)* its non-neighbours are not particularly high compared to their expectations.

In the *kw* and *ksw* settings, the lengthscale parameter $l$ governs whether the metric captures longer or shorter term temporal patterns. For example, if pixel $x_{it}$ and its neighbours increase slowly over time, that change will only cause SPATE to be high for larger lengthscales $l$, while smaller $l$ values imply that current values are compared to those that are close in time. As such, the lengthscale determines what changes are considered "slow" (incorporated into the mean, not detected as space-time interaction) and "fast" (current values are different from the mean, detected as space-time interaction). The *ksw* setting further allows for scenarios where we might wish to compute the metric based on previous time-steps alone, i.e. to preserve sequential logic.

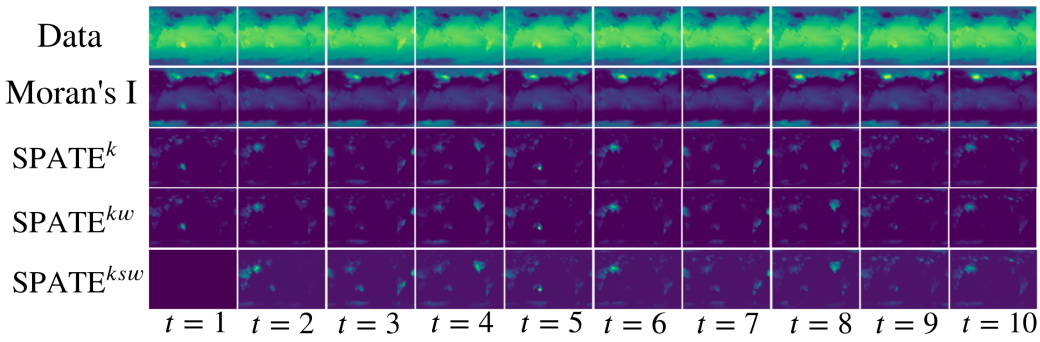



Fig. 5.2 The SPATE metric in its different forms computed for an example from the LGCP datasets. While SPATE$^k$ and SPATE$^{kw}$ are visually undistinguishable, SPATE$^{ksw}$ changes as increasingly more past time steps become available, converging to SPATE$^{kw}$ at time $T$. We can also observe how the Moran's I metric remains static in time, while all versions of SPATE behave dynamically in space and time.

### 5.3.2 SPATE-GAN

Recall that COT can be considered as a maximisation over the classical (Kantorovich) optimal transport (OT) with a temporal causality constraint, which restricts the transporting of mass on the arrival sequence at any time $t$ to depend on the starting sequence only up to time $t$. This motivated us to design the spatio-temporal expectation $\mu_{it}^{ksw}$ in order to respect the nature of sequential data that are generated in an autoregressive manner.

In SPATE-GAN, we integrate our newly devised spatio-temporal metric into the COT-GAN objective function. We compute the embedding for each $x_{it}^d$ and $y_{it}^d$ in minibatches $\{x_{1:T}^d\}_{d=1}^m$ and $\{y_{1:T}^d\}_{d=1}^m$ by

$$\hat{x}_{it}^d = S_{it}(x_{1:T}^d, w) \quad \text{and} \quad \hat{y}_{it}^d = S_{it}(y_{1:T}^d, w),$$

where the binary spatial weight matrix $w$ is pre-defined.

The corresponding embeddings are then concatenated with the training data and generated samples on the channel dimension. We define the empirical measures for the concatenated sequences as

$$\widehat{\mu}^e := \frac{1}{m}\sum_{d=1}^m \delta_{\text{concat}(x_{1:T}^d, \hat{x}_{1:T}^d))},$$
$$\widehat{\nu}_\theta^e := \frac{1}{m}\sum_{d=1}^m \delta_{\text{concat}(y_{1:T}^d, \hat{y}_{1:T}^d)},$$

where concat$(.,.)$ is an operator that concatenates inputs along the channel dimension.

We thus arrive at the objective function for SPATE-GAN:

$$\inf_{\theta}\sup_{\varphi}\left\{\widehat{\mathcal{W}}^{\text{mix}}_{c^{\mathbb{K}}_{\varphi},\varepsilon}(\widehat{\mu}^{e},\widehat{\nu}^{e}_{\theta},\widehat{\mu}^{e'},\widehat{\nu}^{e'}_{\theta}) \quad -\lambda\left[p_{\mathbf{M}_{\varphi_2}}(\widehat{\mu}^{e})+p_{\mathbf{M}_{\varphi_2}}(\widehat{\mu}^{e'})\right]\right\}.$$

We maximise the objective function over $\varphi$ to search for a worst-case distance between the two empirical measures, and minimise it over $\theta$ to learn a distribution that is as close as possible to the real distribution. The algorithm is summarised in Algorithm 3. Its time complexity scales as $\mathcal{O}((J+2n)LTm^2)$ in each iteration where $J$ is the output dimension of the discriminator (see Appendix A for details), and $L$ is the number of Sinkhorn iterations (see Genevay et al. [62], Cuturi [37] for details).

In the experiment section, we will compare SPATE-GAN with three different expectations $\mu^{(k)}_{it}$, $\mu^{(kw)}_{it}$ and $\mu^{(ksw)}_{it}$ in the computation of SPATE. Hence, we denote the corresponding models as SPATE-GAN$^{k}$, SPATE-GAN$^{kw}$, and SPATE-GAN$^{ksw}$, respectively.

Last, we would like to emphasise that, although all three embeddings consider the space-time interactions in a certain way, the non-anticipative assumption of $\mu^{(ksw)}_{it}$ is consistent with the generating process of the type of data we are investigating. As the causality constraint in COT-GAN also restricts the search of transport plans to those that satisfy non-anticipative transporting of mass, SPATE-GAN$^{ksw}$ is a model that fully respects temporal causality in learning, whilst SPATE-GAN$^{k}$ and SPATE-GAN$^{kw}$ also combine information from the future.

## 5.4 Experiments

To empirically evaluate SPATE-GAN, we use three datasets characterised by different spatio-temporal complexities.

**Extreme Weather (EW)** This dataset, introduced by Racah et al. [128], was originally proposed for detecting extreme weather events from a range of climate variables (e.g. zonal winds, radiation). Each of these climate variables is observed four times a day for a $128 \times 192$ pixel representation of the whole earth. We chose to model surface temperatureas it comes with several interesting spatio-temporal characteristics: It exhibits both static (e.g. continent outlines) and dynamic patterns as well as abnormal patterns (e.g. in the presence of tropical cyclones or atmospheric rivers). Furthermore, simulating climate data is an important potential downstream application of deep generative models.

**Algorithm 3:** training SPATE-GAN by SGD

**Data:** $\{x^d_{1:T}\}_{d=1}^n$ (input data), $\zeta$ (latent distribution)
**Parameters:** $\theta_0$, $\varphi_0$ (parameter initialisation), $m$ (batch size), $\varepsilon$ (regularisation parameter), $\alpha$ (learning rate), $\lambda$ (martingale penalty coefficient)
initialise: $\theta \leftarrow \theta_0$, $\varphi \leftarrow \varphi_0$
**for** $b = 1, 2, \ldots$ **do**
  Sample $\{x^d_{1:T}\}_{d=1}^m$ from real data;
  Sample $\{z^d_{1:T}\}_{d=1}^m$ from $\zeta$;
  Generate sequences from latent: $(y^d_{1:T}) \leftarrow g_\theta(z^d_{1:T})$;
  Compute the embeddings: $\hat{x}^d_{it} = S_{it}(x^d_{1:T}, w), \quad \hat{y}^i_{it} = S_{it}(y^i_{1:T}, w)$;
  Concatenated the data with embeddings: $\text{concat}(x^d_{1:T}, \hat{x}^d_{1:T}), \text{concat}(y^d_{1:T}, \hat{y}^d_{1:T})$ ;
  Update discriminator parameter:
  $\varphi \leftarrow \varphi + \alpha \nabla_\varphi \big( \widehat{\mathcal{W}}^{\text{mix}}_{c^{\mathcal{K}}_\varphi, \varepsilon}(\widehat{\mu}^e, \widehat{\nu}^e_\theta, \widehat{\mu}^{e'}, \widehat{\nu}^{e'}_\theta) - \lambda \left[ p_{\mathbf{M}_{\varphi_2}}(\widehat{\mu}^e) + p_{\mathbf{M}_{\varphi_2}}(\widehat{\mu}^{e'}) \right] \big)$;
  Sample $\{z^d_{1:T}\}_{d=1}^m$ from $\zeta$;
  Generate sequences from latent: $(y^d_{1:T}) \leftarrow g_\theta(z^d_{1:T})$;
  Compute the embeddings: $\hat{x}^d_{it} = S_{it}(x^d_{1:T}, w), \quad \hat{y}^d_{it} = S_{it}(y^d_{1:T}, w)$;
  Concatenated the data with embeddings: $\text{concat}(x^d_{1:T}, \hat{x}^d_{1:T}), \text{concat}(y^d_{1:T}, \hat{y}^d_{1:T})$ ;
  Update generator parameter: $\theta \leftarrow \theta - \alpha \nabla_\theta \left( \widehat{\mathcal{W}}^{\text{mix}}_{c^{\mathcal{K}}_\varphi, \varepsilon}(\widehat{\mu}^e, \widehat{\nu}^e_\theta, \widehat{\mu}^{e'}, \widehat{\nu}^{e'}_\theta) \right)$;
**end**

**LGCP** This dataset represents the intensities (number of events in a grid cell) of a log-Gaussian Cox process (LGCP), a continuous spatio-temporal point process. LGCPs are a popular class of models for simulating contagious spatio-temporal patterns and have various applications, for example in epidemiology. We simulate 300 different LGCP intensities on a $64 \times 64$ grid over 10 time steps using the *R* package *LGCP* [156].

**Turbulent Flows (TF)** This dataset, proposed by [175], simulates velocity fields according to the Navier-Stokes equation. This is a class of partial differential equations describing the motion of fluids. Fluid dynamics and simulation is another potential application of deep generative models. Following the approach of Wang et al. [175], we divide the data into 7 steps of $64 \times 64$ pixel frames. Please note that we only utilise the first velocity field, so that all our utilised datasets are single-channel.

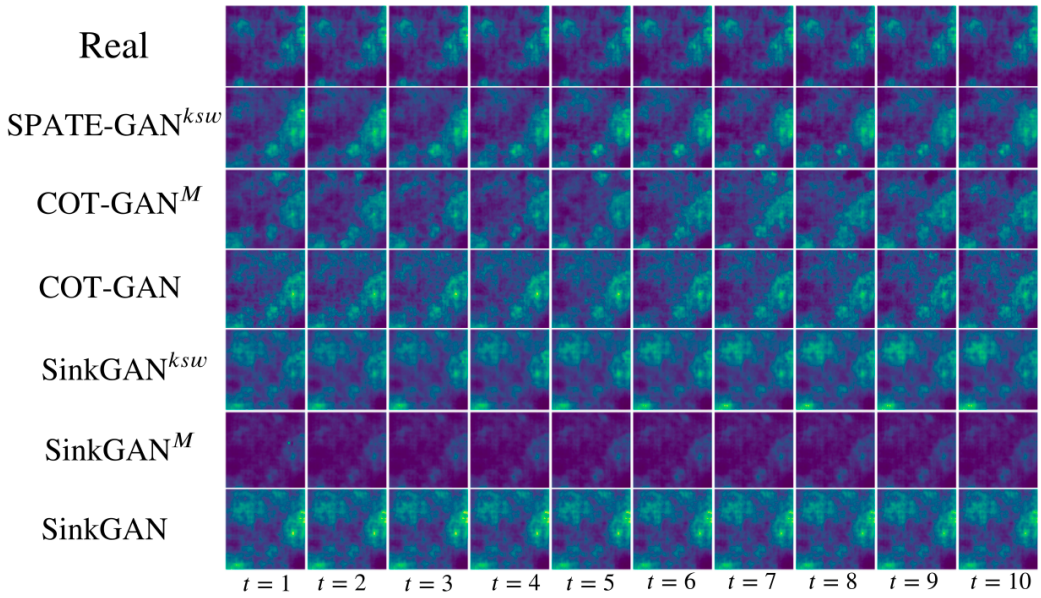



Fig. 5.3 Selected samples for LGCP dataset.

### 5.4.1 Baselines and evaluation metrics

We use COT-GAN [186] and GAN proposed by [62], which we name as SinkGAN, as base models. We augment both models with our new embedding loss, using SPATE with *k*, *kw* and *ksw* configurations. We refer to all models using a COT-GAN backbone in combination with our new embedding loss as SPATE-GAN. We further denote the SinkGAN models corresponding to three SPATE settings as SinkGAN$^{k}$, SinkGAN$^{kw}$ and SinkGAN$^{ksw}$. To compare our approach to a non-time-sensitive embedding, we also deploy models using the Moran's I metric using the same embedding loss procedure, denoted as COT-GAN$^{M}$ and SinkGAN$^{M}$.

To compare our GAN output to real data samples, we use three different metrics: Earth Mover Distance (EMD), Maximum Mean Discrepancy (MMD) [27] and a classifier two-sample test based on a k-nearest-neighbour (KNN) classifier with $k = 1$ [107]. All these measures are general purpose GAN metrics. While GAN metrics specialised on video data exist, they rely on extracting features from models pre-trained on three-channel RGB video data. As we are working with single-channel, non-image data, these methods are not applicable in our case.

### 5.4.2 Experimental Setting

We compare SPATE-GAN to a range of baseline configurations. We use the same GAN architecture for all these settings to ensure comparability. Our GAN generators feed the

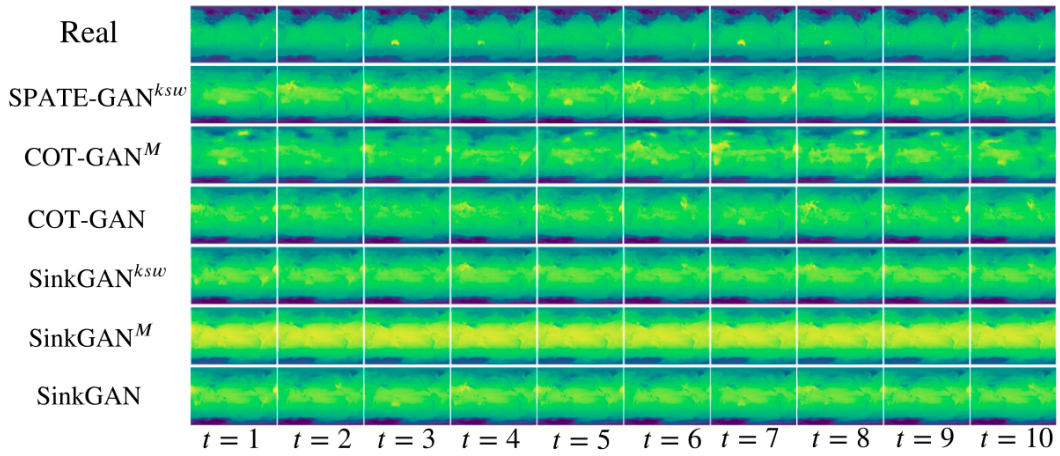



Fig. 5.4 Selected samples for Extreme Weather dataset.

noise input through two LSTM layers to obtain time-dependent features. These are then mapped into the desired shape for deconvolutional operations using a fully-connected layer with a leaky ReLU activation. Lastly, 4 deconvolutional layers map the output into video frames, all also with leaky ReLU activation. Our discriminators initially feed video input through three convolutional layers with leaky ReLU activation. The outputs from the convolutional operations are then reshaped and fed through two LSTM layers to create the final discriminator outputs.

All our models are implemented in PyTorch [124] and optimised using the Adam algorithm [89]. Our experiments are conducted on a single Geforce 1080Ti or RTX 3090 GPU. Further training details can be found in Appendix B.

### 5.4.3 Results

Results from our experiments are shown in Table 5.1. Visual comparisons between real and generated data from the different models are shown in Figures 5.3,5.4, and 5.5. For larger figures including results from all tested model configurations, please see the Appendix D. Through all experiments we can observe that SPATE-GAN$^{ksw}$ consistently outperforms the competing approaches, achieving the best scores across all datasets and evaluation metrics.

This finding is interesting as the *ksw* setting theoretically looses information over the *k* and *kw* approaches, which both have access to future time steps when calculating the SPATE metric. Nevertheless, this result underlines the strong synergies between SPATE$^{ksw}$ and the COT-GAN backbone: The metric is calculated in sequential fashion and thus respects the same causality constraints that restrict COT-GAN. As such, the outcome, while noteworthy, is not surprising.

Table 5.1 Evaluations for LGCP, EW and TF datasets. Lower values in EMD and MMD indicate better sample quality, while values close to 0.5 are more desirable for KNN.

| **LGCP** | EMD | MMD | KNN |
|---|---|---|---|
| SinkGAN | 12.46 (0.02) | 0.38 (0.001) | 0.14 (0.001) |
| SinkGAN$^{M}$ | 12.46 (0.02) | 0.38 (0.001) | 0.14 (0.001) |
| SinkGAN$^{k}$ | 12.65 (0.03) | 0.38 (0.001) | 0.15 (0.001) |
| SinkGAN$^{kw}$ | 10.60 (0.01) | 0.63 (0.008) | 0.30 (0.002) |
| SinkGAN$^{ksw}$ | 13.33 (0.01) | 0.36 (0.001) | 0.38 (0.003) |
| COT-GAN | 12.38 (0.02) | **0.30 (0.001)** | 0.20 (0.004) |
| COT-GAN$^{M}$ | 12.38 (0.02) | **0.30 (0.001)** | 0.20 (0.004) |
| SPATE-GAN$^{k}$ | 11.56 (0.02) | 0.32 (0.01) | 0.31 (0.01) |
| SPATE-GAN$^{kw}$ | 10.92 (0.03) | 0.64 (0.035) | 0.15 (0.006) |
| SPATE-GAN$^{ksw}$ | **10.47 (0.02)** | **0.30 (0.001)** | **0.39 (0.01)** |
| **Extreme Weather** | | | |
| SinkGAN | 29.40 (0.05) | 0.49 (0.001) | 0.41 (0.004) |
| SinkGAN$^{M}$ | 29.27 (0.05) | 0.72 (0.002) | 0.22 (0.01) |
| SinkGAN$^{k}$ | 32.57 (0.03) | 0.81 (0.001) | 0.16 (0.004) |
| SinkGAN$^{kw}$ | 32.78 (0.05) | 0.81 (0.001) | 0.18 (0.004) |
| SinkGAN$^{ksw}$ | 30.00 (0.04) | 0.50 (0.001) | 0.41 (0.004) |
| COT-GAN | 26.66 (0.09) | 0.43 (0.002) | **0.42 (0.002)** |
| COT-GAN$^{M}$ | 36.42 (0.14) | 0.65 (0.002) | 0.09 (0.01) |
| SPATE-GAN$^{k}$ | 33.58 (0.07) | 0.73 (0.002) | 0.15 (0.01) |
| SPATE-GAN$^{kw}$ | 33.36 (0.09) | 0.72 (0.002) | 0.13 (0.003) |
| SPATE-GAN$^{ksw}$ | **26.24 (0.07)** | **0.42 (0.002)** | **0.42 (0.002)** |
| **Turbulent Flows** | | | |
| SinkGAN | 26.52 (0.007) | 1.23 (0.001) | 0.15 (0.001) |
| SinkGAN$^{M}$ | 28.02 (0.005) | 1.22 (0.0002) | 0.01 (0.002) |
| SinkGAN$^{k}$ | 28.14 (0.002) | 1.32 (0.002) | 0.08 (0.001) |
| SinkGAN$^{kw}$ | 30.98 (0.001) | 1.50 (0.001) | 0.03 (0.001) |
| SinkGAN$^{ksw}$ | 25.47 (0.008) | 1.24 (0.0002) | 0.13 (0.002) |
| COT-GAN | 27.03 (0.01) | 1.22 (0.001) | **0.16 (0.002)** |
| COT-GAN$^{M}$ | 24.93 (0.01) | 1.19 (0.001) | 0.09 (0.002) |
| SPATE-GAN$^{k}$ | 25.70 (0.02) | 1.21 (0.001) | 0.12 (0.003) |
| SPATE-GAN$^{kw}$ | 24.30 (0.002) | 1.42 (0.001) | 0.13 (0.004) |
| SPATE-GAN$^{ksw}$ | **22.98 (0.01)** | **1.16 (0.001)** | **0.16 (0.002)** |

This result is strengthened by a comparison with the SinkGAN-based approaches: SinkGAN does not follow the same restrictions and, as we observe, is not improved as consistently by the SPATE-based embedding losses. In fact, in some cases the naive SinkGAN performs better than its derivatives using SPATE or Moran's I based embedding losses.

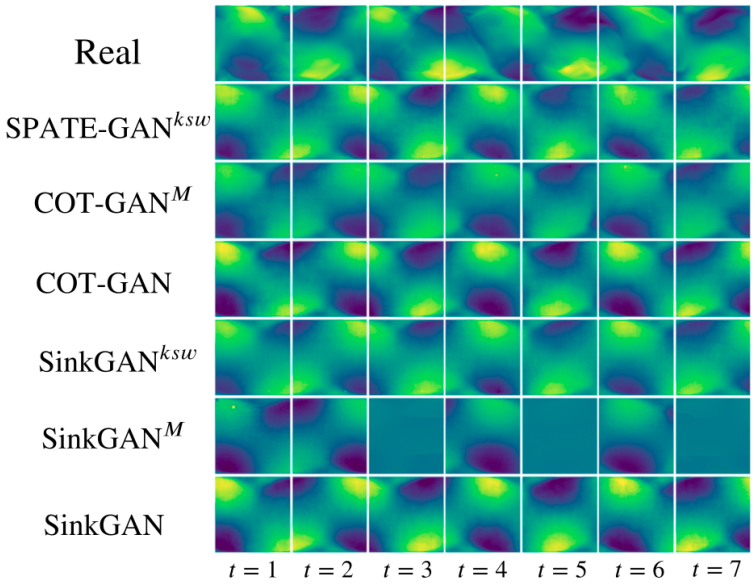



Fig. 5.5 Selected samples for Turbulent Flows dataset.

We also observe that throughout all settings, models using Moran's I perform similarly to their naive counterparts. This confirms that in fact, simply using measures of spatial autocorrelation computed over a sequence is not sufficient for capturing complex spatio-temporal effects. On the contrary, the other two SPATE settings, $k$ and $kw$, both appear to have beneficial effects and improve performance.

In summary, our results highlight how COT-GAN combined with a non-anticipative measure of space-time association can improve the modelling of complex spatio-temporal patterns. This finding represents another step on the way towards deep learning methods specialised on the dynamics driving many systems on our planet.

Furthermore, we provide an investigation on the impact of the lengthscale parameter $l$ in the spatio-temporal expectations for $l \in \{1, 10, 20, 30, 50\}$. As shown in Figure 5.6, $l = 20$ leads to better EMD and KNN results whilst all MMD scores remain unchanged. For the results presented in this chapter, we set $l = 20$ in all our experiments.

## 5.5 Conclusion

Recent studies have called for more research into improving deep learning models for spatio-temporal earth systems data [132]. Other academic domains have dealt with these data for many decades and have developed methods for capturing specific spatial and spatio-temporal effects. Inspired by their approaches, we devise SPATE, a measure of spatio-temporal association capable of detecting emerging space-time clusters and homogeneous areas in the

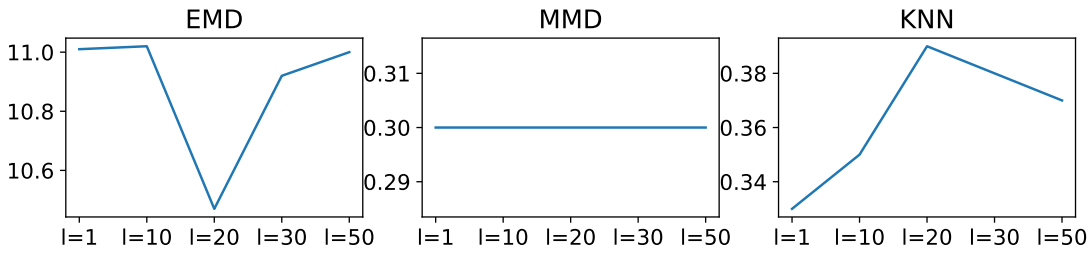



Fig. 5.6 Evaluations of SPATE-GAN$^{ksw}$ (left: EMD, middle: MMD, and right: KNN) on LGCP dataset given lengthscale $l \in \{1, 10, 20, 30, 50\}$.

data. We then develop a novel embedding loss for video GANs utilising SPATE as a means of reinforcing the learning of these patterns-of-interest. Our new generative modelling approach, SPATE-GAN, shows performance increases on a range of different datasets emulating the real-world complexities of spatio-temporal dynamics. As such, this study highlights how domain expertise from applied academic areas can help to motivate methodological advances in machine learning.

# Chapter 6

# Double Generative Adversarial Networks for Conditional Independence Testing

## 6.1 Introduction

Conditional independence (CI) is a fundamental concept in statistics and machine learning. Testing conditional independence is a key building block and plays a central role in a wide variety of statistical learning problems, for instance, causal inference [125], graphical models [96], dimension reduction [102], among others. It is frequently used in a wide range of scientific and business applications, and we demonstrate its application with a cancer genetics example later.

In this chapter, we aim at testing whether two random variables $X$ and $Y$ are conditionally independent given a set of confounding variables $Z$. That is, we test the hypotheses:

$$\mathcal{H}_0 : X \perp\!\!\!\perp Y \mid Z \qquad \text{versus} \qquad \mathcal{H}_1 : X \not\perp\!\!\!\perp Y \mid Z, \tag{6.1.1}$$

given the observed data of $n$ i.i.d. copies $\{(X_i, Y_i, Z_i)\}_{1 \le i \le n}$ of $(X, Y, Z)$. For our problem, $X, Y$ and $Z$ can all be multivariate. However, the main challenge arises when the confounding set of variables $Z$ is high-dimensional. As such, we primarily focus on the scenario with a univariate $X$ and $Y$, and a multivariate $Z$. Meanwhile, our proposed method is applicable to the multivariate $X$ and $Y$ scenario as well. Another challenge is the limited sample size compared to the dimensionality of $Z$. As a result, many existing tests are ineffective, with either an inflated type-I error, or not having enough power to detect the alternatives. See Section 6.2 for a detailed literature review.

To deal with those challenges, we propose a testing procedure based on double generative adversarial networks [GANs, 67] for the CI testing problem in (6.1.1). GANs have

recently stood out as a powerful approach for learning and generating random samples from a complex, high-dimensional data distribution. They have been successfully applied in numerous applications, ranging from image processing and computer vision, to sequential data modelling such as natural language, music, speech, and to medical fields such as DNA design and drug discovery; see [69] for a review of the GANs applications. Moreover, there have recently emerged works studying the consistency and rate of convergence of the GANs estimators; see, e.g., [106, 31].

Our proposal involves two key components: a double GANs framework to learn two generators that approximate the conditional distribution of $X$ given $Z$, and $Y$ given $Z$, respectively, and a test statistic that is taken as the maximum of generalised covariance measures of multiple transformation functions of $X$ and $Y$. We first show that our test statistic is doubly-robust, which offers an additional layer of protection against potential misspecification of the conditional distributions; see Theorems 6.4.1 and 6.4.2. We then show that the resulting test achieves a valid control of the type-I error asymptotically, and more importantly, under the set of conditions that are much weaker and practically more feasible compare to the existing tests; see Theorem 6.4.3. Besides, we prove that the power of our test approaches one asymptotically; see Theorem 6.4.4, and we demonstrate through simulations that it is more powerful than numerous competing tests empirically. In addition, we employ data splitting and cross-fitting that allow us to derive the asymptotic properties under minimal conditions on the generators, and employ multiplier bootstrap to obtain the corresponding $p$-value of the test.

Our contributions are multi-fold. We develop a useful testing procedure for a fundamentally important statistical inference problem. We establish the statistical guarantees under much weaker conditions. We also give an example of how to utilise some state-of-the-art deep learning tools, such as GANs, to address a classical but challenging statistical problem.

The rest of the chapter is organised as follows. Section 6.2 reviews some key existing CI testing methods. Section 6.3 develops the double GANs-based testing procedure. Section 6.4 derives the theoretical properties. Section 6.5 presents the simulations and a cancer genetics data example. Section 6.6 concludes the chapter. The Appendix collects all technical proofs.

## 6.2 Related work

There has been a large and growing literature on conditional independence testing; see Li and Fan [103] for a review. Broadly speaking, the existing tests can be cast into four main categories, the metric-based tests [e.g., 152, 153, 177, 122, 176], the conditional randomisation-based tests [e.g., 29, 19], the kernel-based tests [e.g., 56, 194], and the

regression-based tests [e.g., 77, 143]. There are also some other types of tests [e.g., 21, 22, to name a few].

The metric-based tests typically employ some kernel smoothers to estimate the conditional characteristic function or the distribution function of $Y$ given $X$ and $Z$. Kernel smoothers, however, are known to suffer from the curse of dimensionality, and as such, these tests are usually not suitable when the dimension of $Z$ is high. The conditional randomisation-based tests require the knowledge of the conditional distribution of $X|Z$ [29]. If unknown, the type-I error rates of these tests rely critically on the quality of the approximation of this conditional distribution. Kernel-based tests are built upon the notion of maximum mean discrepancy [MMD, 68], and could have inflated type-I errors. Regression-based tests have valid type-I error control, but may suffer from inadequate power.

Next, we discuss in detail the conditional randomisation-based tests, in particular, the work of [19], the regression-based tests, and the MMD-based tests, as our proposal is related to and built on those methods. For each family of tests, we first lay out the main ideas, then discuss their potential limitations.

### 6.2.1 Conditional randomisation-based tests

The family of conditional randomisation-based tests is built upon the following basis. If the conditional distribution $P_{X|Z}$ of $X$ given $Z$ is known, then one can independently draw $X_i^{(1)} \sim P_{X|Z=Z_i}$, for $i = 1,\dots,n$, where the superscript denotes the first round of draws. Besides, these samples are independent of the observed samples $X_i$'s and $Y_i$'s. Write $\boldsymbol{X} = (X_1,\dots,X_n)^\top$, $\boldsymbol{X}^{(1)} = (X_1^{(1)},\dots,X_n^{(1)})^\top$, $\boldsymbol{Y} = (Y_1,\dots,Y_n)^\top$, and $\boldsymbol{Z} = (Z_1,\dots,Z_n)^\top$. Hereinafter we use boldface letters to denote data matrices that consist of $n$ samples. Since the joint distributions of $(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z})$ and $(\boldsymbol{X}^{(1)},\boldsymbol{Y},\boldsymbol{Z})$ are the same under $\mathcal{H}_0$, any large difference between the two distributions can be interpreted as evidence against $\mathcal{H}_0$. Therefore, one can repeat the sample drawing process $M$ times, i.e., $X_i^{(m)} \sim P_{X|Z=Z_i}$, $i = 1,\dots,n$, $m = 1,\dots,M$. Write $\boldsymbol{X}^{(m)} = (X_1^{(m)},\dots,X_n^{(m)})^\top$. Then, for a given test statistic $\rho = \rho(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z})$, the associated $p$-value is

$$p = \frac{1}{M}\left[\sum_{m=1}^{M} \mathbb{I}\left\{\rho(\boldsymbol{X}^{(m)},\boldsymbol{Y},\boldsymbol{Z}) \geq \rho(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z})\right\}\right],$$

where $\mathbb{I}(\cdot)$ denotes the indicator function. Since the triplets $(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}), (\boldsymbol{X}^{(1)},\boldsymbol{Y},\boldsymbol{Z}),\dots,$ $(\boldsymbol{X}^{(M)},\boldsymbol{Y},\boldsymbol{Z})$ are exchangeable under $\mathcal{H}_0$, the above $p$-value is valid, in the sense that it equals the significance level under the null hypothesis, i.e.,

$$\Pr(p \leq \alpha|\mathcal{H}_0) = \alpha, \quad \text{for any } 0 < \alpha < 1.$$

In practice, however, $P_{X|Z}$ is rarely known. [19] proposed to approximate $P_{X|Z}$ using GANs. Specifically, they learned a generator $\mathbb{G}_X(\cdot,\cdot)$ from the observed data, then took $Z_i$ along with an independent noise variable as the input to obtain a sample $\widetilde{X}_i^{(m)}$, which minimizes the divergence between the distributions of $(X_i, Z_i)$ and $(\widetilde{X}_i^{(m)}, Z_i)$. They computed the $p$-value by replacing $\boldsymbol{X}^{(m)}$ with $\widetilde{\boldsymbol{X}}^{(m)} = (\widetilde{X}_1^{(m)}, \ldots, \widetilde{X}_n^{(m)})^\top$. They called this test the *generative conditional independence test* (GCIT). By Theorem 1 of [19], the excess type-I error of this test is upper bounded as,

$$\begin{aligned} \Pr(p \le \alpha | \mathcal{H}_0) - \alpha &\le \mathrm{E}\left\{ d_{\mathrm{TV}}\left( \widetilde{P}_{\boldsymbol{X}|\boldsymbol{Z}}, P_{\boldsymbol{X}|\boldsymbol{Z}} \right) \right\} \\ &= \mathrm{E}\left\{ \sup_A \left| \Pr(\boldsymbol{X} \in A | \boldsymbol{Z}) - \Pr(\widetilde{\boldsymbol{X}}^{(m)} \in A | \boldsymbol{Z}) \right| \right\} \equiv D, \end{aligned} \tag{6.2.1}$$

where $d_{\mathrm{TV}}$ is the total variation norm between two probability distributions $P$ and $Q$ such that $d_{\mathrm{TV}}(P,Q) = \sup_A |P(A) - Q(A)|$, the supremum is taken over all measurable sets $A$, and the expectations in (6.2.1) are taken with respect to $\boldsymbol{Z}$.

By definition, the error term $D$ in (6.2.1) measures the quality of the conditional distribution approximation. [19] argued that this error term is negligible due to the capacity of deep neural networks in terms of estimating the conditional distribution. To the contrary, we find this approximation error is usually *not* negligible, and consequently, it may inflate the type-I error and invalidate the test. We consider a simple example to further elaborate this.

**Example 6.2.1.** Suppose $X$ is one-dimensional, and follows a simple linear regression model, $X = Z^\top \beta_0 + \varepsilon$, where the error $\varepsilon$ is independent of $Z$, and $\varepsilon \sim N(0, \sigma_0^2)$ for some $\sigma_0^2 > 0$.

Suppose we know a priori that the linear regression model holds. We thus estimate $\beta_0$ by ordinary least squares, and denote the resulting estimator by $\widehat{\beta}$. For simplicity, suppose $\sigma_0^2$ is known too. For this simple example, we have the following result regarding the approximation error $D$.

**Proposition 2.** Suppose the linear regression model holds, the dimension of $Z$ is much smaller than the sample size $n$, and the derived distribution $\widetilde{P}_{\boldsymbol{X}|\boldsymbol{Z}}$ is Normal$(\boldsymbol{Z}\widehat{\beta}, \sigma_0^2 I_n)$, where $I_n$ is the $n \times n$ identity matrix. Then $D$ does not decay to zero.

To facilitate the understanding of the convergence behaviour of $D$, we sketch a few lines of the proof of Proposition 2. The complete proof is given in the Appendix. Let $\widetilde{P}_{X|Z=Z_i}$ denote the conditional distribution of $\widetilde{X}_i^{(m)}$ given $Z_i$, which is Normal$(Z_i^\top \widehat{\beta}, \sigma_0^2)$ in this example. If $D = o(1)$, then,

$$\widetilde{D} \equiv n^{1/2} \sqrt{\mathrm{E}\left\{ d_{\mathrm{TV}}^2 \left( \widetilde{P}_{X|Z=Z_i}, P_{X|Z=Z_i} \right) \right\}} = o(1). \tag{6.2.2}$$

In other words, in order to control the type-I error, GCIT requires the total variation distance measure in (6.2.2) to converge at a faster rate than $n^{-1/2}$. However, this rate cannot be achieved in general. In our Example 1, we have $\widetilde{D} \geq c$ for some constant $c > 0$. Consequently, $D$ in (6.2.1) is not $o(1)$. Proposition 2 shows that, even if we know a priori that the linear model holds, $D$ does not decay to zero as $n$ tends to infinity. In practice, we do not have such prior model information. Then it would be even more difficult to estimate the conditional distribution $P_{X|Z}$. Therefore, using GANs to approximate $P_{X|Z}$ does not guarantee a negligible approximation error.

### 6.2.2 Regression-based tests

The family of regression-based tests is built upon the generalised covariance measure,

$$\mathrm{GCM}(X,Y) = \frac{1}{n}\sum_{i=1}^{n}\left\{X_i - \widehat{\mathrm{E}}(X_i|Z_i)\right\}\left\{Y_i - \widehat{\mathrm{E}}(Y_i|Z_i)\right\},$$

where $\widehat{\mathrm{E}}(X|Z)$ and $\widehat{\mathrm{E}}(Y|Z)$ are the estimated condition means $\mathrm{E}(X|Z)$ and $\mathrm{E}(Y|Z)$, respectively, obtained by some supervised learner. When the prediction errors of $\widehat{\mathrm{E}}(X|Z)$ and $\widehat{\mathrm{E}}(Y|Z)$ satisfy certain convergence rates, [143] proved that GCM is asymptotically normal under $\mathcal{H}_0$, in which the asymptotic mean is zero, and the standard deviation can be consistently estimated by some standard error estimator, denoted by $\widehat{s}(\mathrm{GCM})$. Therefore, at level $\alpha$, we reject $\mathcal{H}_0$, if $|\mathrm{GCM}|/\widehat{s}(\mathrm{GCM})$ exceeds the upper $\alpha/2$th quantile of a standard normal distribution.

Such a test can control the type-I error. Nevertheless, it may not have sufficient power to detect $\mathcal{H}_1$. Consider the asymptotic mean of GCM, which is $\mathrm{GCM}^*(X,Y) = \mathrm{E}\{X - \mathrm{E}(X|Z)\}\{Y - \mathrm{E}(Y|Z)\}$. The regression-based tests require $|\mathrm{GCM}^*|$ to be nonzero under $\mathcal{H}_1$ to have power. However, it may be difficult to satisfy such a requirement. We again consider a simple example.

**Example 6.2.2.** Suppose $X^*$, $Y$ and $Z$ are independent random variables. Besides, $X^*$ has mean zero, and $X = X^* g(Y)$ for some function $g$.

For this example, we have $\mathrm{E}(X|Z) = \mathrm{E}(X)$, since both $X^*$ and $Y$ are independent of $Z$, and so is $X$. Besides, $\mathrm{E}(X) = \mathrm{E}(X^*)\mathrm{E}\{g(Y)\} = 0$, since $X^*$ is independent of $Y$ and $\mathrm{E}(X^*) = 0$. Thus $\mathrm{GCM}^*(X,Y) = \mathrm{E}\{X - \mathrm{E}(X)\}\{Y - \mathrm{E}(Y|Z)\} = 0$ for any function $g$. On the other hand, $X$ and $Y$ are conditionally dependent given $Z$, as long as $g$ is not a constant function. Therefore, for this example, the regression-based tests would fail to discriminate between $\mathcal{H}_0$ and $\mathcal{H}_1$.

### 6.2.3 MMD-based tests

The family of MMD-based tests involves the maximum mean discrepancy as a measure of independence. For any two probability measures $P$, $Q$ and a function space $\mathbb{F}$, define

$$\mathrm{MMD}(P,Q|\mathbb{F}) = \sup_{f\in\mathbb{F}}\{\mathrm{E}f(W_1) - \mathrm{E}f(W_2)\}, \quad \text{where } W_1 \sim P,\ W_2 \sim Q.$$

Let $\mathbb{H}_1$, $\mathbb{H}_2$ denote some function spaces of $X$ and $Y$. Define

$$\phi_{XY} = \mathrm{MMD}(P_{XY}, Q_{XY} \mid \mathbb{H}_1 \otimes \mathbb{H}_2),$$

where $\otimes$ is the tensor product, $P_{XY}$ is the joint distribution of $(X,Y)$ whose definition does not rely on $Z$, and $Q_{XY}$ is the conditionally independent distribution with the same $X$ and $Y$ margins as $P_{XY}$. Let $X'$ and $Y'$ be independent copies of $X$ and $Y$, such that they are conditionally independent given $Z$. Then $Q_{XY}$ corresponds to the joint distribution of $(X',Y')$. Note that, to generate $(X',Y')$, we need to first sample $Z$ according to $P_Z$, then generate $X'$ and $Y'$ that follow $P_{X|Z}$ and $P_{Y|Z}$, respectively. As such, $Q_{XY}$ depends on $Z$, and $\phi_{XY}$ depends on $Z$ through $Q_{XY}$. Furthermore, since $\mathrm{E}\{h_1(X')h_2(Y')\} = \mathrm{E}[\mathrm{E}\{h_1(X')|Z\}\mathrm{E}\{h_2(Y')|Z\}]$, we have,

$$\begin{aligned}
\phi_{XY} &= \sup_{h_1\in\mathbb{H}_1, h_2\in\mathbb{H}_2} \big[\mathrm{E}\{h_1(X)h_2(Y)\} - \mathrm{E}\{h_1(X')h_2(Y')\}\big] \\
&= \sup_{h_1\in\mathbb{H}_1, h_2\in\mathbb{H}_2} \Big(\mathrm{E}\{h_1(X)h_2(Y)\} - \mathrm{E}[\mathrm{E}\{h_1(X)|Z\}\mathrm{E}\{h_2(Y)|Z\}]\Big) \\
&= \sup_{h_1\in\mathbb{H}_1, h_2\in\mathbb{H}_2} \Big(\mathrm{E}\{h_1(X)h_2(Y)\} - \mathrm{E}[h_1(X)\mathrm{E}\{h_2(Y)|Z\}] - \mathrm{E}[\{h_1(X)|Z\}h_2(Y)] \\
&\qquad\qquad + \mathrm{E}[\mathrm{E}\{h_1(X)|Z\}\mathrm{E}\{h_2(Y)|Z\}]\Big) \\
&= \sup_{h_1\in\mathbb{H}_1, h_2\in\mathbb{H}_2} \mathrm{E}\Big[h_1(X) - E\{h_1(X)|Z\}\Big]\Big[h_2(Y) - \mathrm{E}\{h_2(Y)|Z\}\Big].
\end{aligned}$$

As such, $\phi_{XY}$ measures the average conditional association between $X$ and $Y$ given $Z$. Under $\mathcal{H}_0$, it equals zero, and hence an estimator of this measure can be used as a test statistic for $\mathcal{H}_0$. Moreover, if $\mathbb{H}_1$ and $\mathbb{H}_2$ are reproducing kernel Hilbert spaces (RKHSs), then $\phi_{XY}$ has a closed form expression in terms of the reproducing kernels of the RKHS [47, 68], which makes the tests based on an estimator of $\phi_{XY}$ easier to evaluate.

A notable example of this family is the kernel MMD-based test (KCIT) of Zhang et al. [194]. We next further discuss this test. To control the type-I error asymptotically, KCIT requires the dimension $d_Z$ of $Z$ to be fixed [194, Proposition 5], since it uses the continuous mapping theorem to derive the limiting distribution of its test statistic. However, the continu-

ous mapping theorem may not hold when $d_Z$ diverges with $n$. In addition, KCIT requires the $\ell_1$ distance between the covariance operator and its empirical estimator to decay to zero. It remains unknown whether such an assertion holds as $d_Z$ diverges. By contrast, the test we develop allows $d_Z$ to diverge while maintaining the asymptotic control of the type-I error. This implies that our test is expected to have a better size control than KCIT when $d_Z$ is large. We later further verify this through numerical simulations. Moreover, the maximisation of KCIT is done over unit balls in an RKHS, while our proposed test can deal with much more general function spaces such as those generated by neural networks. Consequently, the power of our test can be tailored to more general alternatives than KCIT. For instance, it is known that deep neural networks learn certain non-smooth functions at a faster rate than kernel methods [80]. This implies that our test is expected to have a better power than KCIT under certain types of alternatives.

## 6.3 A new double GANs-based testing procedure

Moreover, to improve the power of the test, we consider a collection of the generalised covariance measures, $\{\mathrm{GCM}(h_1(X), h_2(Y)) : h_1, h_2\}$, for multiple combinations of transformation functions $h_1(X)$ and $h_2(Y)$. We then take the maximum of all these GCMs as our test statistic. This essentially yields a type of maximum mean discrepancy measure $\phi_{XY}$. To see why this statistic can enhance the power, we quickly revisit Example 2. When $g$ is not a constant function, there exists some nonlinear function $h_1$ such that $h_1^*(Y) = \mathrm{E}\{h_1(X)|Y\}$ is not a constant function of $Y$. Set $h_2 = h_1^*$. We then have $\mathrm{GCM}^* = \mathrm{E}[h_1\{X^* g(Y)\}\{Y - \mathrm{E}(Y)\}] = \mathrm{Var}\{h_1^*(Y)\} > 0$, which enables us to discriminate $\mathcal{H}_1$ from $\mathcal{H}_0$.

We note that the maximum of GCMs yields MMD. Instead of using kernels, we have chosen GANs, because they have been shown to give good approximations of complex distributions [80]. This allows the transformation functions $h_1$ and $h_2$ to be arbitrary function spaces. We set these function spaces to the class of neural networks in our implementation. In contrast, kernel based measures such as KCIT are limited to vector spaces of functions, which can be problematic for a high-dimensional conditioning variable [47].

We also remark that, even though our proposal is built upon the existing CI tests, our test is far from a simple extension. The major challenge lies in how to properly utilise the GAN estimators for the purpose of high-dimensional conditional independence testing. Despite the fact that GANs are capable of approximating complex high-dimensional probability distributions, the GAN estimators have non-negligible bias that decays slower than the

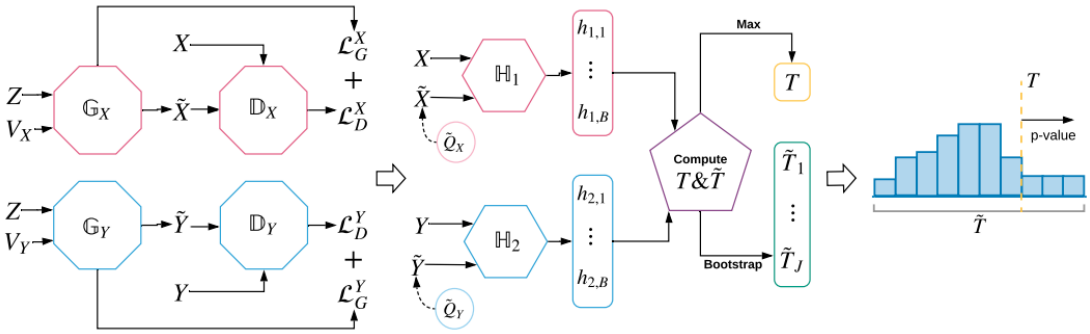



Fig. 6.1 Illustration of the conditional independence test with double GANs.

parametric root-$n$ rate. Naively plugging the GAN estimators in the test statistic can invalidate the subsequent inference.

We give a graphical overview of our proposed testing procedure in Figure 6.1. We first employ double GANs to compute the test statistic that is the maximum of the GCMs over multiple transform functions. We then employ multiplier bootstrap to compute the corresponding $p$-value. We next detail the main components of our testing procedure.

### 6.3.1 Test statistic

We begin with two function spaces, $\mathbb{H}_1 = \{h_{1,\theta_1} : \theta_1 \in \mathbb{R}^{d_1}\}$ and $\mathbb{H}_2 = \{h_{2,\theta_2} : \theta_2 \in \mathbb{R}^{d_2}\}$, indexed by some parameters $\theta_1$ and $\theta_2$, respectively. In our implementation, we set $\mathbb{H}_1$ and $\mathbb{H}_2$ to the classes of neural networks with a single-hidden layer, finitely many hidden nodes, and the sigmoid activation function. However, a broad range of other function spaces may be considered, as appropriate for the application at hand. We next randomly generate $B$ functions, $h_{1,1},\ldots,h_{1,B} \in \mathbb{H}_1$, $h_{2,1},\ldots,h_{2,B} \in \mathbb{H}_2$, where we independently generate i.i.d. multivariate normal variables $\theta_{1,1},\ldots,\theta_{1,B} \sim N(0, 2I_{d_1}/d_1)$, and $\theta_{2,1},\ldots,\theta_{2,B} \sim N(0, 2I_{d_2}/d_2)$. We then set $h_{1,b} = h_{1,\theta_{1,b}}$, and $h_{2,b} = h_{2,\theta_{2,b}}$, $b \in [B] = \{1,\ldots,B\}$. Consider the following maximum-type test statistic,

$$T = \max_{b_1,b_2\in[B]} \widehat{\sigma}^{-1}_{b_1,b_2} \left| \frac{1}{n}\sum_{i=1}^{n} \left[h_{1,b_1}(X_i) - \widehat{\mathrm{E}}\{h_{1,b_1}(X_i)|Z_i\}\right] \left[h_{2,b_2}(Y_i) - \widehat{\mathrm{E}}\{h_{2,b_2}(Y_i)|Z_i\}\right] \right|,$$

where $\widehat{\sigma}^2_{b_1,b_2}$ is the sampling variance estimator,

$$\widehat{\sigma}^2_{b_1,b_2} = \frac{1}{n-1}\sum_{i=1}^{n}\Bigg( \Big[h_{1,b_1}(X_i) - \widehat{\mathrm{E}}\{h_{1,b_1}(X_i)|Z_i\}\Big]\Big[h_{2,b_2}(Y_i) - \widehat{\mathrm{E}}\{h_{2,b_2}(Y_i)|Z_i\}\Big] \\ -\frac{1}{n}\sum_{i=1}^{n}\Big[h_{1,b_1}(X_i) - \widehat{\mathrm{E}}\{h_{1,b_1}(X_i)|Z_i\}\Big]\Big[h_{2,b_2}(Y_i) - \widehat{\mathrm{E}}\{h_{2,b_2}(Y_i)|Z_i\}\Big]\Bigg)^2.$$

To compute $T$, we need to estimate the conditional means, $\mathrm{E}\{h_{1,b_1}(X)|Z\}$ and $\mathrm{E}\{h_{2,b_2}(Y)|Z\}$, which can be done by applying some supervised learning methods. However, this needs to be performed for all $b_1, b_2 \in [B]$. In theory, $B$ should diverge to infinity to guarantee the power property of the test. As such, this approach is computationally very expensive. Instead, we propose to implement this step based on the generators $\mathbb{G}_X$ and $\mathbb{G}_Y$ estimated using GANs, which is much more efficient computationally.

Specifically, we first randomly generate i.i.d. samples $\{v^{(m)}_{i,X}\}_{m=1}^M$, $\{v^{(m)}_{i,Y}\}_{m=1}^M$ from multivariate normal distribution, for $i = 1,\ldots,n$. We then feed $Z_i$ and $v^{(m)}_{i,X}$ into GANs to obtain the pseudo samples $\widetilde{X}^{(m)}_i = \mathbb{G}_X(Z_i, v^{(m)}_{i,X})$, and feed $Z_i$ and $v^{(m)}_{i,Y}$ to obtain $\widetilde{Y}^{(m)}_i = \mathbb{G}_Y(Z_i, v^{(m)}_{i,Y})$, for $i = 1,\ldots,n, m = 1,\ldots,M$. These pseudo samples approximate the conditional distribution of $X_i$ and $Y_i$ given $Z_i$, respectively. We then compute

$$\widehat{\mathrm{E}}\{h_{1,b_1}(\widetilde{X}_i)|Z_i\} = \frac{1}{M}\sum_{m=1}^{M} h_{1,b_1}(\widetilde{X}^{(m)}_i), \quad \widehat{\mathrm{E}}\{h_{2,b_2}(Y_i)|Z_i\} = \frac{1}{M}\sum_{m=1}^{M} h_{2,b_2}(\widetilde{Y}^{(m)}_i),$$

for $b_1, b_2 = 1,\ldots,B$. Plugging the estimated means into $T$ produces the sample test statistic,

$$\widehat{T} = \max_{b_1,b_2}\left| n^{-1/2}\sum_{i=1}^{n}\psi_{b_1,b_2,i}\right|, \quad \text{where} \tag{6.3.1}$$

$$\psi_{b_1,b_2,i} = \widehat{\sigma}^{-1}_{b_1,b_2}\left\{h_{1,b_1}(X_i) - \frac{1}{M}\sum_{m=1}^{M} h_{1,b_1}\left(\widetilde{X}^{(m)}_i\right)\right\}\left\{h_{2,b_2}(Y_i) - \frac{1}{M}\sum_{m=1}^{M} h_{2,b_2}\left(\widetilde{Y}^{(m)}_i\right)\right\}.$$

To help reduce the type-I error, we further employ a data splitting and cross-fitting strategy, which has been commonly used in statistical inferences in recent years [135]. That is, we use different subsets of data samples to learn GANs and to construct the test statistic. We begin by dividing the data into $L$ folds of equal size. We use $\mathcal{I}^{(\ell)}$ to denote the set of indices of subsamples in the $\ell$th fold, and $\mathcal{I}^{(-\ell)}$ its complement. We next learn two generators $\mathbb{G}^{(\ell)}_X$ and $\mathbb{G}^{(\ell)}_Y$, based on $\{(X_i,Z_i)\}_{i\in\mathcal{I}^{(-\ell)}}$ and $\{(Y_i,Z_i)\}_{i\in\mathcal{I}^{(-\ell)}}$, to approximate the conditional distributions of $X|Z$ and $Y|Z$, for $\ell = 1,\cdots,L$. Finally, for each $\ell$ and $i \in \mathcal{I}^{(\ell)}$, we generate the pseudo samples $\widetilde{X}^{(m)}_i$ and $\widetilde{Y}^{(m)}_i$ using $\mathbb{G}^{(\ell)}_X$ and $\mathbb{G}^{(\ell)}_Y$, and construct $\widehat{T}$ as in (6.3.1). In this

**Algorithm 4:** Algorithm for computing the test statistic.

**Input:** The number of transformation functions $B$, the number of pseudo samples $M$, and the number of data splits $L$.

**Step 1:** Divide $\{1,\ldots,n\}$ into $L$ folds $\mathcal{I}^{(1)},\ldots,\mathcal{I}^{(L)}$. Denote $\mathcal{I}^{(-\ell)} = \{1,\ldots,n\}\backslash\mathcal{I}^{(\ell)}$.

**Step 2:** For $\ell = 1,\ldots,L$, train two generators $\mathbb{G}_X^{(\ell)}$ and $\mathbb{G}_Y^{(\ell)}$ based on $\{(X_i,Z_i)\}_{i\in\mathcal{I}^{(-\ell)}}$ and $\{(Y_i,Z_i)\}_{i\in\mathcal{I}^{(-\ell)}}$, to approximate the conditional distributions of $X|Z$ and $Y|Z$.

**Step 3:** For $\ell = 1,\ldots,L$ and $i \in \mathcal{I}_\ell$, generate i.i.d. random noises $\left\{v_{i,X}^{(m)}\right\}_{m=1}^{M}$, $\left\{v_{i,Y}^{(m)}\right\}_{m=1}^{M}$. Set $\widetilde{X}_i^{(m)} = \mathbb{G}_X^{(\ell)}\left(Z_i, v_{i,X}^{(m)}\right)$, and $\widetilde{Y}_i^{(m)} = \mathbb{G}_Y^{(\ell)}\left(Z_i, v_{i,Y}^{(m)}\right)$, $m = 1,\ldots,M$.

**Step 4:** Randomly generate $h_{1,1},\ldots,h_{1,B} \in \mathbb{H}_1$ and $h_{2,1},\ldots,h_{2,B} \in \mathbb{H}_2$.

**Step 5:** Compute the test statistic $\widehat{T}$.

way, $\widetilde{X}_i^{(m)}$ and $\widetilde{Y}_i^{(m)}$ are conditionally independent of the observations in $\mathcal{I}^{(\ell)}$ given $Z_i$. Such a cross-fitting strategy allows us to derive the asymptotic properties of the test under minimal conditions on the generators.

We summarise our procedure of computing the test statistic in Algorithm 4.

### 6.3.2 Approximation of conditional distribution via GANs

There are numerous GANs methods available for learning high-dimensional distributions. We adopt the proposal of [61] to learn the conditional distributions $P_{X|Z}$ and $P_{Y|Z}$ in our setting thanks to its competitive performance. Recall that $\widetilde{P}_{X|Z}$ is the distribution of pseudo outcome generated by the generator $\mathbb{G}_X$ given $Z$. We consider estimating $P_{X|Z}$ by optimising

$$\min_{\mathbb{G}_X}\max_{c}\widetilde{\mathcal{D}}_{c,\varepsilon}(P_{X|Z},\widetilde{P}_{X|Z}).$$

Here $\widetilde{\mathcal{D}}_{c,\varepsilon}$ denotes the Sinkhorn loss function between two probability measures with respect to some cost function $c$ and some regularisation parameter $\varepsilon > 0$,

$$\begin{aligned}\widetilde{\mathcal{D}}_{c,\varepsilon}(\mu,\nu) &= 2\mathcal{D}_{c,\varepsilon}(\mu,\nu) - \mathcal{D}_{c,\varepsilon}(\mu,\mu) - \mathcal{D}_{c,\varepsilon}(\nu,\nu),\\ \mathcal{D}_{c,\varepsilon}(\mu,\nu) &= \inf_{\pi\in\Pi(\mu,\nu)}\int_{x,y}\left\{c(x,y) - \varepsilon H(\pi|\mu\otimes\nu)\right\}\pi(dx,dy),\end{aligned}$$

where $\Pi(\mu,\nu)$ is a set containing all probability measures $\pi$ whose marginal distributions correspond to $\mu$ and $\nu$, $H$ is the Kullback-Leibler divergence, and $\mu\otimes\nu$ is the product measure of $\mu$ and $\nu$. When $\varepsilon = 0$, $\mathcal{D}_{c,0}(\mu,\nu)$ measures the optimal transport of $\mu$ into $\nu$ with respect to the cost function $c(\cdot,\cdot)$ [37]. When $\varepsilon \neq 0$, an entropic regularisation is added to this optimal transport. As such, the objective function $\mathcal{D}_{c,\varepsilon}$ is a regularised optimal transport

metric, and the regularisation is to facilitate the computation, so that $\mathcal{D}_{c,\varepsilon}$ can be efficiently evaluated.

Intuitively, the closer the two probability measures, the smaller the Sinkhorn loss. As such, maximising the loss with respect to the cost function learns a discriminator that can better discriminate the samples generated between $P_{X|Z}$ and $\widetilde{P}_{X|Z}$. On the other hand, minimising the maximum cost with respect to the generator $\mathbb{G}_X$ makes it closer to the true distribution $P_{X|Z}$. This yields the minimax formulation $\min_{\mathbb{G}_X} \max_c \widetilde{\mathcal{D}}_{c,\varepsilon}(P_{X|Z}, \widetilde{P}_{X|Z})$ that we target. In practice, we approximate the cost and the generator based on neural networks. Integrations in the objective function $\widetilde{\mathcal{D}}_{c,\varepsilon}(P_{X|Z}, \widetilde{P}_{X|Z})$ are approximated by sample averages. The conditional distribution of $P_{Y|Z}$ is estimated similarly.

### 6.3.3 Bootstrap for the $p$-value

Next, we propose a multiplier bootstrap method to approximate the distribution of $\widehat{T}$ under $\mathcal{H}_0$ and compute the corresponding $p$-value. Let $\psi_{b_1,b_2} = n^{-1}\sum_{i=1}^n \psi_{b_1,b_2,i}$. The key observation is that $\{\psi_{b_1,b_2}\}_{b_1,b_2=1}^B$ are asymptotically multivariate normal with zero mean under $\mathcal{H}_0$; see the proof of Theorem 6.4.3 for details. Consequently, $\widehat{T} = \max_{b_1,b_2} |n^{-1/2}\sum_{i=1}^n \psi_{b_1,b_2,i}|$ is to converge to a maximum of normal variables in absolute values.

To approximate this limiting distribution, we first estimate the covariance matrix of a $B^2$-dimensional vector formed by $\{n^{-1/2}\psi_{b_1,b_2}\}_{b_1,b_2=1}^B$ using the sample covariance matrix $\widehat{\Sigma}$, whose $\{b_1+B(b_2-1), b_3+B(b_4-1)\}$th entry is given by

$$\frac{1}{n}\sum_{i=1}^n (\psi_{b_1,b_2,i} - \psi_{b_1,b_2})(\psi_{b_3,b_4,i} - \psi_{b_3,b_4}), \quad b_1,b_2,b_3,b_4 = 1,\ldots,B.$$

We then generate i.i.d. random vectors with the covariance matrix equal to $\widehat{\Sigma}$. This can be achieved by generating i.i.d. standard normal variables $\{W_{i,j}\}_{i,j}$ for $1 \le i \cdots \le n$ and $j = 1,\cdots,J$, then compute $B^2$-dimensional normal random vectors $\boldsymbol{W}_j$ whose $\{b_1+B(b_2-1)\}$th entry is given by $n^{-1/2}\sum_{i=1}^n(\psi_{b_1,b_2,i} - \psi_{b_1,b_2})W_{i,j}$ for $j = 1,\cdots,J$. We next compute $\widetilde{T}_j = \|\boldsymbol{W}_j\|_\infty$, for $j = 1,\ldots,J$, where $\|\cdot\|_\infty$ is the maximum element of a vector in absolute value, and $J$ is the number of bootstrap samples. Finally, we use these maximum absolute values to approximate the distribution of $\widehat{T}$ under the null hypothesis. This yields the $p$-value, $p = J^{-1}\sum_{j=1}^J \mathbb{I}(\widehat{T} \ge \widetilde{T}_j)$. We summarise this bootstrap procedure in Algorithm 5.

## 6.4 Asymptotic theory

To derive the theoretical properties of the test statistic $\widehat{T}$, we first introduce the concept of the "oracle" test statistic $T^*$. If $P_{X|Z}$ and $P_{Y|Z}$ were known a priori, then one can draw $\{X_i^{(m)}\}_m$ and $\{Y_i^{(m)}\}_m$ from $P_{X|Z=Z_i}$ and $P_{Y|Z=Z_i}$ directly, and can compute the test statistic by replacing $\{\widetilde{X}_i^{(m)}\}_m$ and $\{\widetilde{Y}_i^{(m)}\}_m$ with $\{X_i^{(m)}\}_m$ and $\{Y_i^{(m)}\}_m$. We call the resulting $T^*$ an "oracle" test statistic. We next establish the double-robustness property of $\widehat{T}$, which helps explain why our test can relax the requirement in (6.2.2). Roughly speaking, the double-robustness means that $\widehat{T}$ is asymptotically equivalent to $T^*$ when either the conditional distribution of $X|Z$, or that of $Y|Z$, is well approximated by GANs. It guarantees that $\widehat{T}$ converges to $T^*$ at a faster rate than the estimated conditional distribution. In contrast, the convergence rate of the GCIT test statistic is the same as the rate of the estimated conditional distribution. For this reason, our procedure only requires a weaker condition.

**Theorem 6.4.1** (Double-robustness)**.** Suppose $M$ is proportional to $n$, and $B = O(n^c)$ for some constant $c > 0$. Suppose $\min_{h_1 \in \mathbb{H}_1, h_2 \in \mathbb{H}_2} \mathrm{Var}[\{h_1(X) - \mathrm{E}\{h_1(X)|Z\}\}\{h_2(Y) - \mathrm{E}\{h_2(Y)|Z\}\}] \geq c^*$ for some constant $c^* > 0$. Then, $\widehat{T} - T^* = o_p(1)$, when

$$\mathrm{E}\left[d_{\mathrm{TV}}^2\left\{\widetilde{Q}_X^{(\ell)}(\cdot|Z), Q_X(\cdot|Z)\right\}\right] = o(\log^{-1} n), \text{ or } \mathrm{E}\left[d_{\mathrm{TV}}^2\left\{\widetilde{Q}_Y^{(\ell)}(\cdot|Z), Q_Y(\cdot|Z)\right\}\right] = o(\log^{-1} n).$$

We note that the conditions on $M$ and $B$ are mild, as these are user-specified parameters. As we have mentioned, when both total variation distances converge to zero, the test statistic $T$ converges at a faster rate than those total variation distances. Therefore, we can greatly relax the condition in (6.2.2), and replace it with,

$$\left[\mathrm{E}\left\{d_{\mathrm{TV}}^2\left(\widetilde{P}_{X|Z}^{(\ell)}, P_{X|Z}\right)\right\}\right]^{1/2} = O(n^{-\kappa_x}), \text{ and } \left[\mathrm{E}\left\{d_{\mathrm{TV}}^2\left(\widetilde{P}_{Y|Z}^{(\ell)}, P_{Y|Z}\right)\right\}\right]^{1/2} = O(n^{-\kappa_y}), \quad (6.4.1)$$

for some constants $0 < \kappa_x, \kappa_y < 1/2$ and any $\ell \in [L]$, where $\widetilde{P}_{X|Z}^{(\ell)}$ and $\widetilde{P}_{Y|Z}^{(\ell)}$ denote the conditional distributions approximated via GANs trained on the $\ell$-th subset of data samples. The next theorem summarises this discussion.

**Algorithm 5:** Algorithm for computing the $p$-value.

**Input:** The number of bootstrap samples $J$, and $\{\psi_{b_1,b_2,i}\}_{b_1,b_2=1,i=1}^{B,n}$.
**Step 1:** Generate i.i.d. standard normal variables $W_{i,j}$ for $i = 1, \cdots, n$, $j = 1, \ldots, J$.
**Step 2:** Compute $B^2$-dimensional normal random vectors $\boldsymbol{W}_j$ whose $\{b_1 + B(b_2 - 1)\}$th entry is given by $n^{-1/2}\sum_{i=1}^n(\psi_{b_1,b_2,i} - \psi_{b_1,b_2})W_{i,j}$ and set $\widetilde{T}_j = \|\boldsymbol{W}_j\|_\infty$ for $j = 1, \cdots, J$.
**Step 3:** Compute the $p$-value, $p = J^{-1}\sum_{j=1}^J \mathbb{I}(\widehat{T} \geq \widetilde{T}_j)$.

**Theorem 6.4.2.** Suppose the conditions in Theorem 6.4.1. Furthermore, suppose (6.4.1) holds. Then, $\widehat{T} - T^* = O_p\left(n^{-(\kappa_x+\kappa_y)}\log n\right)$.

Since $\kappa_x, \kappa_y > 0$, the convergence rate of $(\widehat{T} - T^*)$ is faster than that in (6.4.1). To ensure $\sqrt{n}(T - T^*) = o_p(1)$, it suffices to require $\kappa_x + \kappa_y > 1/2$. In contrast to (6.2.2), this rate is achievable. We consider two examples in [22] to illustrate this, while the condition holds in a much wider range of settings.

**Example 6.4.1** (Parametric setting)**.** Suppose the parametric forms of $Q_X$ and $Q_Y$ are correctly specified. Then under certain regularity conditions, the requirement $\kappa_x + \kappa_y > 1/2$ holds if $k_x = O(n^{t_x})$ and $k_y = O(n^{t_y})$ for some $t_x + t_y < 1/2$, where $k_x$ and $k_y$ are the dimensions of the parameters defining the parametric models for $Q_X$ and $Q_y$, respectively.

**Example 6.4.2** (Nonparametric setting with binary data)**.** Suppose $X, Y$ are binary variables. Then the requirement $\kappa_x + \kappa_y > 1/2$ holds if the mean squared prediction errors of the nonparametric estimators of the conditional means of $X$ and $Y$ given $Z$ are $O(n^{-t_x})$ and $O(n^{-t_y})$ for some $t_x$, $t_y$, such that $t_x + t_y > 1/2$.

We briefly remark that, there is no explicit specification on $d_Z$ in the statement of Theorem 6.4.2. It is implicitly imposed due to the requirement that $\kappa_x + \kappa_y > 1/2$, and $d_Z$ is allowed to diverge with the sample size. In addition, the condition $\kappa_x + \kappa_y > 1/2$ can be further relaxed to $\kappa_1, \kappa_2 > 0$ using the theory of higher order influence functions [**? ? ?** ]. However, the resulting estimators would be considerably much more complicated, and thus we do not pursue those estimators.

Next, we show that our proposed test can control the type-I error asymptotically.

**Theorem 6.4.3.** Suppose the conditions in Theorem 6.4.1 hold. Suppose (6.4.1) holds for some $\kappa_x$, $\kappa_y$ such that $\kappa_x + \kappa_y > 1/2$. Then, the $p$-value from Algorithm 5 satisfies that $\Pr(p \le \alpha | \mathcal{H}_0) = \alpha + o(1)$.

Next, to derive the asymptotic power of the test, we introduce the pair of hypotheses based on the notion of weak conditional independence [39],

$$\begin{aligned}
&\mathcal{H}_0^* : \mathrm{E}[\mathrm{cov}\{f(X), g(Y)|Z\}] = 0, \quad \text{for any } f \in L_X^2, g \in L_Y^2 \quad \text{versus} \\
&\mathcal{H}_1^* : \mathrm{E}[\mathrm{cov}\{f(X), g(Y)|Z\}] \neq 0, \quad \text{for some } f \in L_X^2, g \in L_Y^2,
\end{aligned}$$

where $L_X^2$ and $L_Y^2$ denote the class of all squared integrable functions of $X$ and $Y$, respectively. We note that conditional independence implies weak conditional independence, i.e., $\mathcal{H}_0$ implies $\mathcal{H}_0^*$, and $\mathcal{H}_1^*$ implies $\mathcal{H}_1$. We consider an example to further elaborate on the difference between weak CI and CI.

**Example 6.4.3.** Let $X, Y, Z$ be binary random variables with the distribution functions,

$$\begin{pmatrix} \Pr(X=0,Y=0|Z=0) & \Pr(X=0,Y=1|Z=0) \\ \Pr(X=1,Y=0|Z=0) & \Pr(X=1,Y=1|Z=0) \end{pmatrix} = \begin{pmatrix} 1/6 & 1/3 \\ 1/3 & 1/6 \end{pmatrix},$$
$$\begin{pmatrix} \Pr(X=0,Y=0|Z=1) & \Pr(X=0,Y=1|Z=1) \\ \Pr(X=1,Y=0|Z=1) & \Pr(X=1,Y=1|Z=1) \end{pmatrix} = \begin{pmatrix} 1/3 & 1/6 \\ 1/6 & 1/3 \end{pmatrix},$$

and $Z$ takes the value $\{0,1\}$ with equal probability. We can show that, for any $x, y \in \{0,1\}$,

$$\mathrm{E}\{\Pr(X=x|Z)\Pr(Y=y|Z)\} = \frac{1}{2}\times\frac{1}{2} = \frac{1}{4},$$
$$\begin{aligned}\Pr(X=x,Y=y) &= \frac{1}{2}\Big\{\Pr(X=x,Y=y|Z=0)+\Pr(X=x,Y=y|Z=1)\Big\} \\ &= \frac{1}{2}\times\left(\frac{1}{6}+\frac{1}{3}\right) = \frac{1}{4}.\end{aligned}$$

By definition, this implies that $X$ and $Y$ are weakly conditionally independent given $Z$, since

$$\begin{aligned}\mathrm{E}[\mathrm{cov}\{f(X),g(Y)|Z\}] = \sum_{x,y} f(x)g(y)\Big\{&\Pr(X=x,Y=y) \\ &- \mathrm{E}\big\{\Pr(X=x|Z)\Pr(Y=y|Z)\big\}\Big\} = 0.\end{aligned}$$

However, $\Pr(X=0,Y=0|Z=0) \neq \Pr(X=0|Z=0)\Pr(Y=0|Z=0)$, since the former equals $1/6$, and the latter equals $1/4$. As such, $X$ and $Y$ are not conditionally independent given $Z$.

The next theorem shows that our proposed test is consistent against the alternatives in $\mathcal{H}_1^*$, but not against all alternatives in $\mathcal{H}_1$.

**Theorem 6.4.4.** Suppose the conditions in Theorem 6.4.3 hold, $B = c_0 n^c$ for some $c_0, c > 0$, and $X$, $Y$ are bounded random variables. Then the $p$-value from Algorithm 5 satisfies that $\Pr(p \leq \alpha | \mathcal{H}_1^*) \to 1$, as $n \to \infty$.

Finally, we remark that our test is constructed based on $\phi_{XY}$. Meanwhile, we may consider another test based on $\phi_{XYZ} = \mathrm{MMD}(P_{XYZ}, Q_{XYZ}|\mathbb{H}_1 \otimes \mathbb{H}_2 \otimes \mathbb{H}_3)$, where $P_{XYZ}$ is the joint distribution of $(X,Y,Z)$, $Q_{XYZ} = P_{X|Z}P_{Y|Z}P_Z$, and $\mathbb{H}_3$ is a neural network class of functions of $Z$. This type of test is consistent against all alternatives in $\mathcal{H}_1$. However, in our numerical experiments, we find it less powerful compared to our test. This agrees with the observation by [103] in that, even though the tests based on weak CI cannot fully summarise CI, they potentially enjoy an improved power.

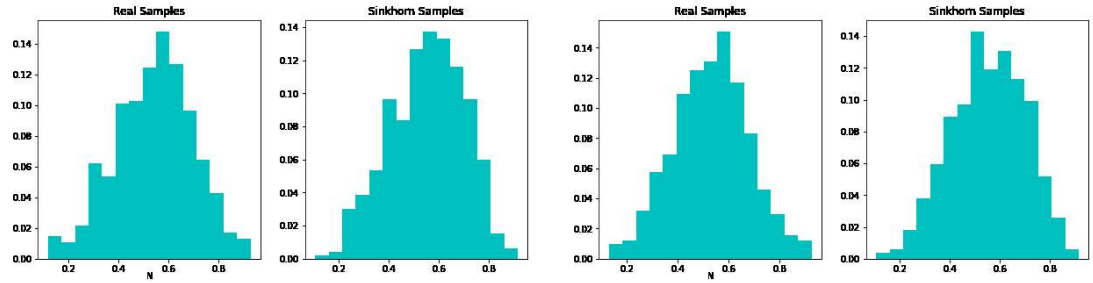
## 6.5 Numerical studies

We begin with a discussion of some implementation details. We then carry out simulations to study the empirical size and power of the proposed test, and compare with several alternative methods. We further illustrate with an application to a cancer genetics example.

### 6.5.1 Implementation details

For the number of functions $B$ in Algorithm 6.4.1, it represents a trade-off. By Theorem 6.4.4, $B$ should be as large as possible to guarantee a good power. In practice, the computation complexity increases as $B$ increases. Our numerical studies suggest that the value of $B$ between 30 and 50 achieves a good balance between the power and the computational cost, and we fix $B = 30$. For the number of pseudo samples $M$, and the number of sample splittings $L$, we find the results are not overly sensitive to their choices, and thus we fix $M = 100$ and $L = 3$. Besides, we set the number of bootstrap samples $J = 1000$.

For the GANs, we use a single-hidden layer neural network to approximate both the discriminator and the generator. The number of nodes in the hidden layer is set at 128. The dimension of the input noise $v_{i,X}^{(m)}$ and $v_{i,Y}^{(m)}$ is set at 10. These tuning parameters are chosen following the common practice in the GANs literature, and also by investigating the goodness-of-fit of the resulting generator, which can be done by comparing the conditional histogram of the generated samples to that of the true samples. In our experiments, we find such an approach yields GANs with satisfactory performances. More specifically, let $d_Z$ denote the dimension of $Z$, and $\widehat{\mu}_Z$ the sample average $n^{-1}\sum_i Z_i$. Let $\widetilde{Y}_i = G_Y(Z_i, v_{i,Y})$ denote a simulated sample to approximate the distribution of $Y|Z = Z_i$ obtained by the generator $G_Y$. When $G_Y$ is accurate, we expect the conditional distribution of $\widetilde{Y}_i$ and $Y_i$ given $Z_i$ are similar. As such, for any $d_Z$-dimensional vector $a$, the histograms $\{\widetilde{Y}_i : a^\top(\widetilde{Z}_i - \widehat{\mu}_Z) > 0\}$ and $\{Y_i : a^\top(Z_i - \widehat{\mu}_Z) > 0\}$ should be similar. We sample i.i.d. vectors $\{a_g\}_g$ from Normal$(0, I_{d_Z})$. For each $g$, we plot the histogram $\{Y_i : a_g^\top(Z_i - \widehat{\mu}_Z) > 0\}$ and $\{\widetilde{Y}_i^{(m)} : a_g^\top(Z_i - \widehat{\mu}_Z) > 0\}$. See Figures 6.2 (a) and (b) for the conditional histograms with two choices of $a_g$. It is seen that the GANs fit the conditional density reasonably well. The fitted conditional distribution for $P_{X|Z}$ can be checked in a similar fashion.




(a) One random value of $a_g$ (b) Another random value of $a_g$

Fig. 6.2 Conditional histograms. GANs are trained using data generated from the simulation study in Section 6.5.2.

### 6.5.2 Simulations

We generate the data following the post nonlinear noise model similarly as in [194, 47, 19], i.e.,

$$X = \sin(a_f^\top Z + \varepsilon_f), \quad \text{and} \quad Y = \cos(a_g^\top Z + bX + \varepsilon_g).$$

The entries of $a_f, a_g$ are randomly and uniformly sampled from $[0,1]$, then normalized to the unit $\ell_1$ norm. The noise variables $\varepsilon_f, \varepsilon_g$ are independently sampled from a normal distribution with mean zero and variance 0.25. In this model, the parameter $b$ determines the degree of conditional dependence. When $b = 0$, $\mathcal{H}_0$ holds, and otherwise $\mathcal{H}_1$ holds. The sample size is set at $n = 1000$.

We call our test DGCIT, short for double GANs-based conditional independence test. We compare it with the GCIT test of [19], the regression-based test (RCIT) of [143], the kernel MMD-based test (KCIT) of [194], and the classifier CI test (CCIT) of [142].

We first study the empirical size when $b = 0$. We vary the dimension of $Z$ as $d_Z = 50, 100, 150, 200, 250$, and consider two generation distributions. We first generate $Z$ from a standard normal distribution, then from a Laplace distribution. We set the significance level at $\alpha = 0.05$ and 0.1. Figure 6.3 reports the empirical size of the tests aggregated over 500 data replications. We make the following observations. First, the type-I error rates of our test and RCIT are close to or below the nominal level in nearly all cases. Second, KCIT fails in that its type-I error is considerably larger than the nominal level in all cases. We suspect it is due to the high-dimensional setting where $d_Z \geq 50$. We have experimented with $d_Z = 5$, and found that KCIT is able to control the type-I error in that case. This is consistent with Proposition 5 of [194], which suggests that KCIT should work in a low-dimensional setting. Third, GCIT and CCIT both have inflated type-I errors in some cases. Take GCIT

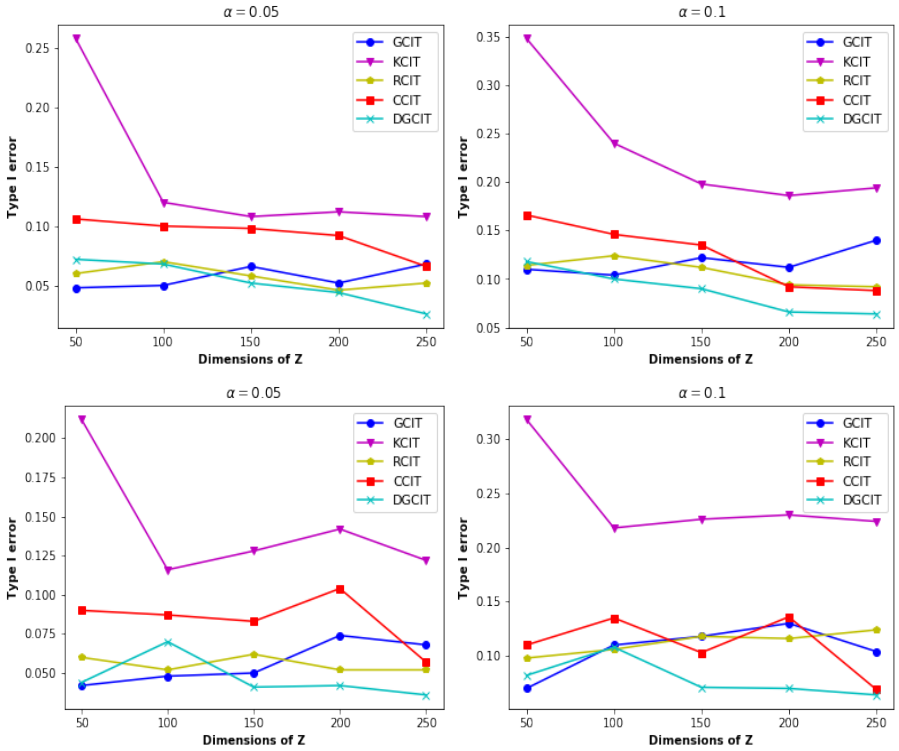



Fig. 6.3 The empirical type-I error rate of various tests under $\mathcal{H}_0$. Left panels: $\alpha = 0.05$, right panels: $\alpha = 0.1$. Top panels: $Z$ is normal, bottom panels: $Z$ is Laplacian.

as an example. When $Z$ is normal, $d_Z = 250$ and $\alpha = 0.1$, its empirical size is close to 0.15. This is consistent with our discussion in Section 6.2.1, since GCIT requires a rather strong condition to control the type-I error.

We then study the empirical power when $b > 0$. We generate $Z$ from a standard normal distribution, with $d_Z = 100, 200$, and vary the value of $b = 0.3, 0.45, 0.6, 0.75, 0.9$ that controls the magnitude of the alternative. Figure 6.4 reports the empirical power of the tests over 500 data replications. We observe that our test is the most powerful, and the empirical power approaches 1 as $b$ increases to 0.9, demonstrating the consistency of the test. Meanwhile, both GCIT and RCIT have no power in all cases. We do not report the power of KCIT, because as we have shown earlier, it cannot control the size, and thus its empirical power is not meaningful.

Finally, we discuss the computation time. All experiments were run on a 16 N1 CPUs Google Cloud Computing platform. The wall clock time for running the entire GCIT test for one data replication was about 2.5 minutes. In contrast, the running time for CCIT was about 2 minutes, for KCIT about 30 seconds, and for GCIT and RCIT about 20 seconds.

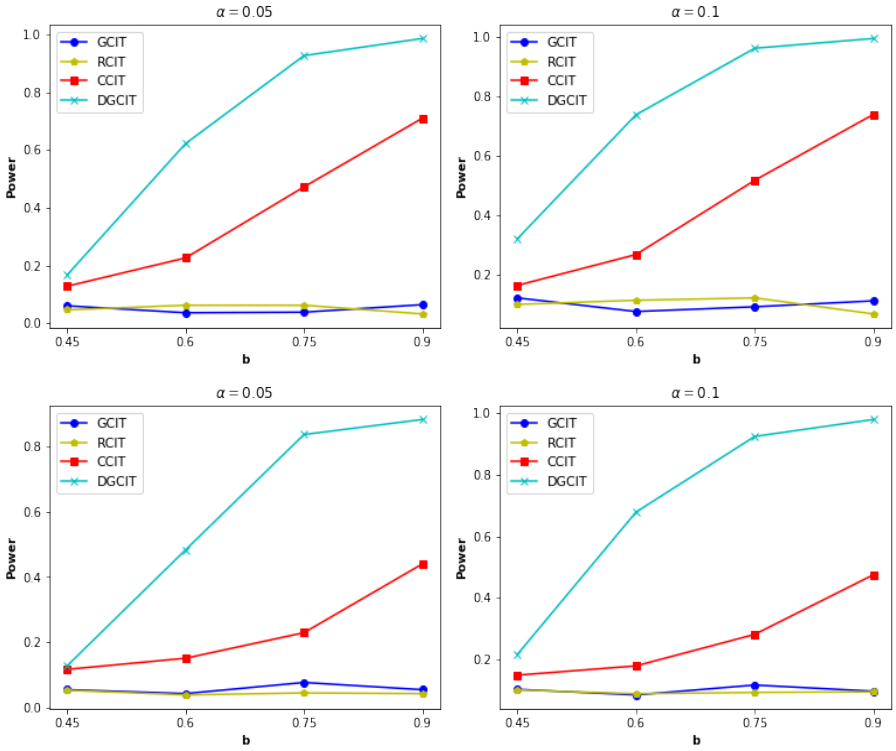



Fig. 6.4 The empirical power of various tests under $\mathcal{H}_1$. Left panels: $\alpha = 0.05$, right panels: $\alpha = 0.1$. Top panels: $d_Z = 100$, bottom panels: $d_Z = 200$.

### 6.5.3 Anti-cancer drug data example

We illustrate our proposed test with an anti-cancer drug dataset from the Cancer Cell Line Encyclopedia (CCLE) [17]. We concentrate on a subset, the CCLE data, that measures the treatment response of drug PLX4720. It is well known that the patient's cancer treatment response to drug can be strongly influenced by alterations in the genome [59]. This data measures 1638 genetic mutations of $n = 472$ cell lines, and the goal of our analysis is to determine which genetic mutation is significantly correlated with the drug response after conditioning on all other mutations. The same data was also analysed in [155] and [19]. We adopt the same screening procedure as theirs to screen out irrelevant mutations, which leaves a total of 466 potential mutations for our conditional independence testing.

The ground truth is unknown for this data. Instead, we compare with the variable importance measures obtained from fitting an elastic net (EN) model and a random forest (RF) model as reported in [17]. In addition, we compare with the GCIT test of [19]. Table 6.1 reports the corresponding variable importance measures and the $p$-values, for 10 mutations that were also reported by [19]. We see that, the $p$-values of the tests generally agree well with the variable important measures from the EN and RF models. Meanwhile, the two conditional

Table 6.1 The variable importance measures of the elastic net and random forest models, versus the $p$-values of the GCIT and DGCIT tests for the anti-cancer drug example.

| | BRAF.V600E | BRAF.MC | HIP1 | FTL3 | CDC42BPA | THBS3 | DNMT1 | PRKD1 | PIP5K1A | MAP3K5 |
|---|---|---|---|---|---|---|---|---|---|---|
| EN | 1 | 3 | 4 | 5 | 7 | 8 | 9 | 10 | 19 | 78 |
| RF | 1 | 2 | 3 | 14 | 8 | 34 | 28 | 18 | 7 | 9 |
| GCIT | <0.001 | <0.001 | 0.008 | 0.521 | 0.050 | 0.013 | 0.020 | 0.002 | 0.001 | <0.001 |
| DGCIT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.794 |

independence tests agree relatively well, except for two genetic mutations, MAP3K5 and FTL3. GCIT concluded that MAP3K5 is significant ($p < 0.001$) but FTL3 is not ($p = 0.521$), whereas our test leads to the opposite conclusion that MAP3K5 is insignificant ($p = 0.794$) but FTL3 is ($p = 0$). Besides, both EN and RF place FTL3 as an important mutation. We then compare our findings with the cancer drug response literature. Actually, MAP3K5 has not been previously reported in the literature as being directly linked to the PLX4720 drug response. Meanwhile, there is strong evidence showing the connections of the FLT3 mutation with cancer response [161, 98]. Combining the existing literature with our theoretical and synthetic results, we have more confidence about the findings of our proposed test.

## 6.6 Discussion

In this chapter, we have developed a new inferential procedure for high-dimensional conditional independence testing, where the dimension of the conditional variables can diverge with the sample size. Our proposal utilises a set of state-of-the-art deep learning tools to help address a classical statistics and machine learning problem. It integrates GANs, neural networks, cross-fitting and multiplier bootstrap. It achieves the asymptotic guarantees under much weaker conditions, and enjoys better empirical performances, when compared to the existing tests. As a tradeoff, our test is computationally more complicated. Nevertheless, the wall clock time for running the entire test for one data replication is in the order of a few minutes and is deemed reasonable. Finally, the computer code is publicly available on the GitHub repository: https://github.com/tianlinxu312/dgcit.

# Chapter 7

# Future Research

In this thesis, we reviewed the most representative deep generative models from the probabilistic modelling perspective, explained the concept of Optimal Transport and its application in generative modelling, and introduced the theory of Causal Optimal Transport (COT) which enforces a causality constraint on the transport plans. In the presented works, we demonstrated the applicability of COT for various tasks in machine learning. The performance of algorithms built on COT suggests that constraining the transport plans to be causal is a promising direction for learning sequential data.

The results of COT-GAN in Chapter 3 (see also [187]) and KCCOT-GAN in Chapter 4 (see also [185]) indicate that the models are capable of capturing both the spatial and temporal features for video generation and prediction. Furthermore, SPATE-GAN in Chapter 5 (see also [95]) shows that performance increases on a range of different datasets emulating the real-world complexities of spatio-temporal dynamics can be achieved by leveraging human expert knowledge. In addition, Chapter 6 (see also [144]) shows how to utilise the state-of-the-art generative models to help address the classical but challenging statistical problem of conditional independence testing.

As a generic metric, COT has a wide range of potential applications. For example, the conditional version of COT-GAN can be used to predict the movement of stock prices. Whilst the noise in video data can be negligible, financial data typically requires special care before modelling in order to achieve reliable predictions. In particular, neural networks are sensitive to outliers which need to be dealt with in the stage of data pre-processing. Another potential application of COT is the field of natural language processing. It is known that a word in a sentence can be determined by not only those that appear before but also those that come after. For this reason, some state-of-the-art language models assume bi-directional correlation in the sequences of language data, see e.g. [170, 42]. An interesting direction for

future research would be to utilise COT to capture the correlation from both directions of the sequences.

# References


[1] Acciaio, B., Backhoff-Veraguas, J., and Carmona, R. (2019a). Extended mean field control problems: stochastic maximum principle and transport perspective. *SIAM Journal on Control and Optimization*, 57(6).

[2] Acciaio, B., Backhoff-Veraguas, J., and Jia, J. (2020). Cournot-nash equilibrium and optimal transport in a dynamic setting. *arXiv preprint arXiv:2002.08786*.

[3] Acciaio, B., Backhoff-Veraguas, J., and Zalashko, A. (2019b). Causal optimal transport and its links to enlargement of filtrations and continuous-time stochastic optimization. *Stochastic Processes and their Applications*.

[4] Aigner, S. and Körner, M. (2018). Futuregan: Anticipating the future frames of video sequences using spatio-temporal 3d convolutions in progressively growing gans. *arXiv preprint arXiv:1810.01325*.

[5] Alain, G., Bengio, Y., Yao, L., Yosinski, J., Thibodeau-Laufer, E., Zhang, S., and Vincent, P. (2016). Gsns: generative stochastic networks. *Information and Inference: A Journal of the IMA*, 5(2):210–249.

[6] Anselin, L. (1995). Local Indicators of Spatial Association—LISA. *Geographical Analysis*, 27(2):93–115.

[7] Aodha, O. M., Cole, E., and Perona, P. (2019). Presence-only geographical priors for fine-grained image classification. In *Proceedings of the IEEE International Conference on Computer Vision*.

[8] Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein gan. *arXiv preprint arXiv:1701.07875*.

[9] Babaeizadeh, M., Finn, C., Erhan, D., Campbell, R. H., and Levine, S. (2017). Stochastic variational video prediction. *ICLR*.

[10] Backhoff, J., Bartl, D., Beiglböck, M., and Wiesel, J. (2020a). Estimating processes in adapted Wasserstein distance. *arXiv preprint arXiv:2002.07261*.

[11] Backhoff, J., Bartl, D., Beiglböck, M., and Wiesel, J. (2020b). Estimating processes in adapted Wasserstein distance. *arXiv preprint arXiv:2002.07261*.

[12] Backhoff, J., Beiglbock, M., Lin, Y., and Zalashko, A. (2017a). Causal transport in discrete time and applications. *SIAM Journal on Optimization*, 27(4).

[13] Backhoff, J., Beiglbock, M., Lin, Y., and Zalashko, A. (2017b). Causal transport in discrete time and applications. *SIAM Journal on Optimization*, 27(4):2528–2562.

[14] Backhoff-Veraguas, J., Bartl, D., Beiglböck, M., and Eder, M. (2020). Adapted Wasserstein distances and stability in mathematical finance. *Finance and Stochastics*, 24(3):601–632.

[15] Bailer, C., Varanasi, K., and Stricker, D. (2017). CNN-based patch matching for optical flow with thresholded hinge embedding loss. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-Janua, pages 2710–2719.

[16] Bao, H., Zhou, X., Zhang, Y., Li, Y., and Xie, Y. (2020). COVID-GAN: Estimating Human Mobility Responses to COVID-19 Pandemic through Spatio-Temporal Conditional Generative Adversarial Networks. In *SIGSPATIAL: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, pages 273–282, New York, NY, USA. Association for Computing Machinery.

[17] Barretina, J., Caponigro, G., Stransky, N., Venkatesan, K., Margolin, A. A., Kim, S., Wilson, C. J., Lehár, J., Kryukov, G. V., Sonkin, D., et al. (2012). The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483(7391):603–607.

[18] Bellemare, M. G., Danihelka, I., Dabney, W., Mohamed, S., Lakshminarayanan, B., Hoyer, S., and Munos, R. (2017). The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*.

[19] Bellot, A. and van der Schaar, M. (2019). Conditional independence testing using generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 2199–2208.

[20] Bengio, Y. and Bengio, S. (2000). Modeling high-dimensional discrete data with multi-layer neural networks. In *Advances in Neural Information Processing Systems 12: Proceedings of the 1999 Conference*, volume 1, page 400. MIT Press.

[21] Bergsma, W. P. (2004). *Testing conditional independence for continuous random variables*. Eurandom.

[22] Berrett, T. B., Wang, Y., Barber, R. F., and Samworth, R. J. (2019). The conditional permutation test for independence while controlling for confounders. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, accepted.

[23] Bińkowski, M., Sutherland, D. J., Arbel, M., and Gretton, A. (2018). Demystifying MMD GANs. In *ICLR*.

[24] Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

[25] Blank, M., Gorelick, L., Shechtman, E., Irani, M., and Basri, R. (2005). Actions as space-time shapes. In *ICCV*.

[26] Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877.

[27] Borgwardt, K. M., Gretton, A., Rasch, M. J., Kriegel, H. P., Schölkopf, B., and Smola, A. J. (2006). Integrating structured biological data by Kernel Maximum Mean Discrepancy. In *Bioinformatics*.

[28] Brix, A. and Diggle, P. J. (2001). Spatiotemporal prediction for log-Gaussian Cox processes. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 63(4):823–841.

[29] Candes, E., Fan, Y., Janson, L., and Lv, J. (2018). Panning for gold:'model-x'knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(3):551–577.

[30] Castrejon, L., Ballas, N., and Courville, A. (2019). Improved conditional vrnns for video prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7608–7617.

[31] Chen, M., Liao, W., Zha, H., and Zhao, T. (2020a). Statistical guarantees of generative adversarial networks for distribution estimation. *arXiv preprint arXiv:2002.03938*.

[32] Chen, N., Zhang, Y., Zen, H., Weiss, R. J., Norouzi, M., and Chan, W. (2020b). Wavegrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713*.

[33] Chernozhukov, V., Chetverikov, D., and Kato, K. (2017). Detailed proof of nazarov's inequality. *arXiv preprint arXiv:1711.10696*.

[34] Chu, G., Potetz, B., Wang, W., Howard, A., Song, Y., Brucher, F., Leung, T., Adam, H., and Research, G. (2019). Geo-Aware Networks for Fine-Grained Recognition. In *International Conference on Computer Vision (ICCV)*, pages 0–0.

[35] Clark, A., Donahue, J., and Simonyan, K. (2019a). Adversarial video generation on complex datasets. *arXiv preprint arXiv:1907.06571*.

[36] Clark, A., Donahue, J., and Simonyan, K. (2019b). Efficient video generation on complex datasets.

[37] Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300.

[38] Das, M. and Ghosh, S. K. (2017). Measuring Moran's I in a cost-efficient manner to describe a land-cover change pattern in large-scale remote sensing imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(6):2631–2639.

[39] Daudin, J. (1980). Partial association measures and an application to qualitative regression. *Biometrika*, 67(3):581–590.

[40] De Cao, N., Aziz, W., and Titov, I. (2020). Block neural autoregressive flow. In *Uncertainty in artificial intelligence*, pages 1263–1273. PMLR.

[41] Denton, E. and Fergus, R. (2018). Stochastic video generation with a learned prior. In *International Conference on Machine Learning*, pages 1174–1183. PMLR.

[42] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[43] Devroye, L., Mehrabian, A., and Reddad, T. (2018). The total variation distance between high-dimensional gaussians. *arXiv preprint arXiv:1810.08693*.

[44] Diggle, P. J. and Gratton, R. J. (1984). Monte carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(2):193–212.

[45] Dinh, L., Krueger, D., and Bengio, Y. (2014). Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*.

[46] Donahue, C., McAuley, J. J., and Puckette, M. S. (2019). Adversarial audio synthesis. In *ICLR*.

[47] Doran, G., Muandet, K., Zhang, K., and Schölkopf, B. (2014). A permutation-based kernel conditional independence test. In *UAI*, pages 132–141.

[48] Doretto, G., Chiuso, A., Wu, Y. N., and Soatto, S. (2003). Dynamic textures. *International Journal of Computer Vision*, 51(2).

[49] Dudley, R. M. (2018). *Real analysis and probability*. CRC Press.

[50] Egiazarian, K., Astola, J., Ponomarenko, N., Lukin, V., Battisti, F., and Carli, M. (2006). New full-reference quality metrics based on hvs. In *Proceedings of the second international workshop on video processing and quality metrics*, volume 4.

[51] Eslami, S. A., Rezende, D. J., Besse, F., Viola, F., Morcos, A. S., Garnelo, M., Ruderman, A., Rusu, A. A., Danihelka, I., Gregor, K., et al. (2018). Neural scene representation and rendering. *Science*, 360(6394):1204–1210.

[52] Esteban, C., Hyland, S. L., and Rätsch, G. (2017). Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633*.

[53] Filntisis, P. P., Efthymiou, N., Potamianos, G., and Maragos, P. (2020). Emotion Understanding in Videos Through Body, Context, and Visual-Semantic Embedding Loss. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 12535 LNCS, pages 747–755. Springer Science and Business Media Deutschland GmbH.

[54] Fournier, N. and Guillin, A. (2015). On the rate of convergence in Wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3):707–738.

[55] Frey, B. J. (1998). *Graphical models for machine learning and digital communication*. MIT press.

[56] Fukumizu, K., Gretton, A., Sun, X., and Schölkopf, B. (2008). Kernel measures of conditional dependence. In *Advances in neural information processing systems*, pages 489–496.

[57] Gaffke, N. and Rüscherndorf, L. (1981). On a class of extremal problems in statistics. *Mathematische Operationsforschung und Statistik. Series Optimization*, 12(1):123–135.

[58] Gao, Y., Cheng, J., Meng, H., and Liu, Y. (2019). Measuring spatio-temporal autocorrelation in time series data of collective human mobility. *Geo-Spatial Information Science*, 22(3):166–173.

[59] Garnett, M., Edelman, E., Gill, S., Greenman, C., Dastur, A., Lau, K., Greninger, P., Thompson, R., Luo, X., Soares, J., Liu, Q., Iorio, F., Surdez, D., Chen, L., Milano, R., Bignell, G., Tam, A., Davies, H., Stevenson, J., and Benes, C. (2012). Systematic identification of genomic markers of drug sensitivity in cancer cells. *Nature*, 483:570–5.

[60] Genevay, A., Chizat, L., Bach, F., Cuturi, M., and Peyré, G. (2019). Sample complexity of sinkhorn divergences. In *AISTATS*.

[61] Genevay, A., Peyré, G., and Cuturi, M. (2017). Learning generative models with sinkhorn divergences. *arXiv preprint arXiv:1706.00292*.

[62] Genevay, A., Peyre, G., and Cuturi, M. (2018). Learning generative models with sinkhorn divergences. In *AISTATS*.

[63] Geng, X., Li, Y., Wang, L., Zhang, L., Yang, Q., Ye, J., and Liu, Y. (2019). Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, volume 33, pages 3656–3663. AAAI Press.

[64] Getreuer, P. (2013). A survey of gaussian convolution algorithms. *Image Processing On Line*, 2013:286–310.

[65] Ghafoorian, M., Nugteren, C., Baka, N., Booij, O., and Hofmann, M. (2019). EL-GAN: Embedding loss driven generative adversarial networks for lane detection. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11129 LNCS, pages 256–272.

[66] Good, I. J. (1963). Maximum entropy for hypothesis formulation, especially for multidimensional contingency tables. *The Annals of Mathematical Statistics*, 34(3):911–934.

[67] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

[68] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773.

[69] Gui, J., Sun, Z., Wen, Y., Tao, D., and Ye, J. (2020). A review on generative adversarial networks: Algorithms, theory, and applications. *arXiv preprint arXiv:2001.06937*.

[70] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777.

[71] Haddad, R. A., Akansu, A. N., et al. (1991). A class of fast gaussian binomial filters for speech and image processing. *IEEE Transactions on Signal Processing*, 39(3):723–727.

[72] Haradal, S., Hayashi, H., and Uchida, S. (2018). Biosignal data augmentation based on generative adversarial networks. In *International Conference in Medicine and Biology Society*.

[73] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*.

[74] Hinton, G. E. and Sejnowski, T. J. (1983). Optimal perceptual inference. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, volume 448, pages 448–453. Citeseer.

[75] Hiriart-Urruty, J.-B. and Lemaréchal, C. (2004). *Fundamentals of convex analysis*. Springer Science & Business Media.

[76] Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851.

[77] Hoyer, P. O., Janzing, D., Mooij, J. M., Peters, J., and Schölkopf, B. (2009). Nonlinear causal discovery with additive noise models. In *Advances in neural information processing systems*, pages 689–696.

[78] Huang, C.-W., Krueger, D., Lacoste, A., and Courville, A. (2018). Neural autoregressive flows. In *International Conference on Machine Learning*, pages 2078–2087. PMLR.

[79] Huang, L., Zhuang, J., Cheng, X., Xu, R., and Ma, H. (2021). STI-GAN: Multimodal Pedestrian Trajectory Prediction Using Spatiotemporal Interactions and a Generative Adversarial Network. *IEEE Access*, 9:50846–50856.

[80] Imaizumi, M. and Fukumizu, K. (2019). Deep neural networks learn non-smooth functions effectively. In *The 22nd international conference on artificial intelligence and statistics*, pages 869–878. PMLR.

[81] Jia, J. and Benson, A. R. (2020). Residual Correlation in Graph Neural Network Regression. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 588–598, New York, NY, USA. Association for Computing Machinery.

[82] Kalchbrenner, N., Oord, A., Simonyan, K., Danihelka, I., Vinyals, O., Graves, A., and Kavukcuoglu, K. (2017). Video pixel networks. In *International Conference on Machine Learning*, pages 1771–1779. PMLR.

[83] Kantorovich, L. V. (2006). On the translocation of masses. *Journal of mathematical sciences*, 133(4):1381–1382.

[84] Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

[85] Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018). Progressive growing of GANs for improved quality, stability, and variation. *ICLR*.

[86] Kim, J., Lee, K., Lee, D., Jin, S. Y., and Park, N. (2020a). DPM: A Novel Training Method for Physics-Informed Neural Networks in Extrapolation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9):8146–8154.

[87] Kim, S. Y., Oh, J., and Kim, M. (2020b). JSI-GAN: GAN-based joint super-resolution and inverse tone-mapping with pixel-wise task-specific filters for UHD HDR video. In *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, volume 34, pages 11287–11295. AAAI press.

[88] Kim, T., Ahn, S., and Bengio, Y. (2019). Variational temporal abstraction. *NeurIPS*.

[89] Kingma, D. P. and Ba, J. L. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

[90] Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. (2016). Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29.

[91] Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[92] Klemmer, K., Koshiyama, A., and Flennerhag, S. (2019). Augmenting correlation structures in spatial data using deep generative models. *arXiv:1905.09796*.

[93] Klemmer, K. and Neill, D. B. (2021). Auxiliary-task learning for geographic data with autoregressive embeddings. In *SIGSPATIAL: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*.

[94] Klemmer, K., Saha, S., Kahl, M., Xu, T., and Zhu, X. X. (2021a). Generative modeling of spatio-temporal weather patterns with extreme event conditioning. *ICLR'21 Workshop AI: Modeling Oceans and Climate Change (AIMOCC)*.

[95] Klemmer, K., Xu, T., Acciaio, B., and Neill, D. B. (2021b). Spate-gan: Improved generative modeling of dynamic spatio-temporal patterns with an autoregressive embedding loss. *arXiv preprint arXiv:2109.15044*.

[96] Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. Adaptive computation and machine learning. MIT Press.

[97] Kulldorff, M., Heffernan, R., Hartman, J., Assunção, R., and Mostashari, F. (2005). A space-time permutation scan statistic for disease outbreak detection. *PLoS Medicine*, 2(3):0216–0224.

[98] Larrosa-Garcia, M. and Baer, M. R. (2017). Flt3 inhibitors in acute myeloid leukemia: Current status and future directions. *Molecular Cancer Therapeutics*, 16(6):991–1001.

[99] LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. (2006). A tutorial on energy-based learning. *Predicting structured data*, 1(0).

[100] Lee, A. X., Zhang, R., Ebert, F., Abbeel, P., Finn, C., and Levine, S. (2018). Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*.

[101] Lee, J. and Li, S. (2017). Extending Moran's Index for Measuring Spatiotemporal Clustering of Geographic Events. *Geographical Analysis*, 49(1):36–57.

[102] Li, B. (2018). *Sufficient Dimension Reduction: Methods and Applications with R*. CRC Press.

[103] Li, C. and Fan, X. (2019). On nonparametric conditional independence tests for continuous variables. *Wiley Interdisciplinary Reviews: Computational Statistics*, page e1489.

[104] Li, C.-L., Chang, W.-C., Cheng, Y., Yang, Y., and Póczos, B. (2017). Mmd gan: Towards deeper understanding of moment matching network. In *NeurIPS*.

[105] Li, Y. and Mandt, S. (2018). Disentangled sequential autoencoder. In *ICML*, pages 5670–5679.

[106] Liang, T. (2018). On how well generative adversarial networks learn densities: Nonparametric and parametric results. *arXiv preprint arXiv:1811.03179*.

[107] Lopez-Paz, D. and Oquab, M. (2019). Revisiting classifier two-sample tests. In *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*.

[108] Mai, G., Janowicz, K., Yan, B., Zhu, R., Cai, L., and Lao, N. (2020). Multi-Scale Representation Learning for Spatial Feature Distributions using Grid Cells. In *International Conference on Learning Representations (ICLR)*.

[109] Marshall, A. W. and Olkin, I. (1968). Scaling of matrices to achieve specified row and column sums. *Numerische Mathematik*, 12(1):83–90.

[110] Mathieu, M., Couprie, C., and LeCun, Y. (2016). Deep multi-scale video prediction beyond mean square error. *ICLR*.

[111] Matthews, J. L., Diawara, N., and Waller, L. A. (2019). Quantifying Spatio-Temporal Characteristics via Moran's Statistics. In *STEAM-H: Science, Technology, Engineering, Agriculture, Mathematics and Health*, pages 163–177. Springer Nature.

[112] Mogren, O. (2016). C-rnn-gan: A continuous recurrent neural network with adversarial training. In *Constructive Machine Learning Workshop (CML) at NIPS*.

[113] Mohamed, S. and Lakshminarayanan, B. (2016). Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*.

[114] Monge, G. (1781). Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*.

[115] Müller, A. (1997). Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443.

[116] Nguyen, X., Wainwright, M. J., and Jordan, M. I. (2010). Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861.

[117] Nixon, M. and Aguado, A. (2019). *Feature extraction and image processing for computer vision*. Academic press.

[118] Nowozin, S., Cseke, B., and Tomioka, R. (2016). f-gan: Training generative neural samplers using variational divergence minimization. In *NeurIPS*.

[119] Oh, J., Guo, X., Lee, H., Lewis, R., and Singh, S. (2015). Action-conditional video prediction using deep networks in atari games. *NIPS*.

[120] Oord, A. v. d., Vinyals, O., and Kavukcuoglu, K. (2017). Neural discrete representation learning. *NeurIPS*.

[121] Ord, J. K. and Getis, A. (2012). Local spatial heteroscedasticity (LOSH). *Annals of Regional Science*, 48(2):529–539.

[122] Pan, W., Wang, X., Wen, C., Styner, M., and Zhu, H. (2017). Conditional local distance correlation for manifold-valued data. In *International Conference on Information Processing in Medical Imaging*, pages 41–52. Springer.

[123] Papamakarios, G., Pavlakou, T., and Murray, I. (2017). Masked autoregressive flow for density estimation. *arXiv preprint arXiv:1705.07057*.

[124] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32.

[125] Pearl, J. (2009). *Causality: Models, Reasoning and Inference*. Cambridge: Cambridge University Press, 2nd Edition.

[126] Pflug, G. C. and Pichler, A. (2012). A distance for multistage stochastic optimization models. *SIAM Journal on Optimization*, 22(1):1–23.

[127] Pflug, G. C. and Pichler, A. (2016). From empirical observations to tree models for stochastic optimization: convergence properties. *SIAM Journal on Optimization*, 26(3):1715–1740.

[128] Racah, E., Beckham, C., Maharaj, T., Kahou, S. E., Prabhat, and Pal, C. (2017). ExtremeWeather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. In *Advances in Neural Information Processing Systems*, volume 2017-Decem, pages 3403–3414.

[129] Rao, C., Sun, H., and Liu, Y. (2020). Physics-informed deep learning for incompressible laminar flows. *Theoretical and Applied Mechanics Letters*, 10(3):207–212.

[130] Reed, S., Oord, A., Kalchbrenner, N., Colmenarejo, S. G., Wang, Z., Chen, Y., Belov, D., and Freitas, N. (2017). Parallel multiscale autoregressive density estimation. In *International Conference on Machine Learning*, pages 2912–2921. PMLR.

[131] Reed, S. E., Zhang, Y., Zhang, Y., and Lee, H. (2015). Deep visual analogy-making. In *NeurIPS*.

[132] Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., and Prabhat (2019). Deep learning and process understanding for data-driven Earth system science. *Nature*, 566(7743):195–204.

[133] Reinhard, E., Heidrich, W., Debevec, P., Pattanaik, S., Ward, G., and Myszkowski, K. (2010). *High dynamic range imaging: acquisition, display, and image-based lighting*. Morgan Kaufmann.

[134] Rickart, C. (1943). Decomposition of additive set functions. *Duke Mathematical Journal*, 10(4):653–665.

[135] Romano, J. and DiCiccio, C. (2019). Multiple data splitting for testing. Technical report, Technical report.

[136] Saito, M., Matsumoto, E., and Saito, S. (2017). Temporal generative adversarial nets with singular value clipping. In *ICCV*.

[137] Sajjadi, M. S., Bachem, O., Lucic, M., Bousquet, O., and Gelly, S. (2018). Assessing generative models via precision and recall. *Advances in neural information processing systems*, 31.

[138] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans. *Advances in neural information processing systems*, 29.

[139] Salimans, T. and Ho, J. (2022). Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*.

[140] Salimans, T., Karpathy, A., Chen, X., and Kingma, D. P. (2017). Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*.

[141] Salimans, T., Zhang, H., Radford, A., and Metaxas, D. (2018). Improving GANs using optimal transport. In *ICLR*.

[142] Sen, R., Suresh, A. T., Shanmugam, K., Dimakis, A. G., and Shakkottai, S. (2017). Model-powered conditional independence test. In *Advances in neural information processing systems*, pages 2951–2961.

[143] Shah, R. D. and Peters, J. (2018). The hardness of conditional independence testing and the generalised covariance measure. *arXiv preprint arXiv:1804.07203*.

[144] Shi, C., Xu, T., Bergsma, W., and Li, L. (2021). Double generative adversarial networks for conditional independence testing. *Journal of Machine Learning Research*, 22(285):1–32.

[145] Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *arXiv preprint arXiv:1506.04214*.

[146] Siino, M., Rodríguez-Cortés, F. J., Mateu, J., and Adelfio, G. (2018). Testing for local structure in spatiotemporal point pattern data. In *Environmetrics*, volume 29, page e2463. John Wiley and Sons Ltd.

[147] Sinkhorn, R. and Knopp, P. (1967). Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348.

[148] Sismanidis, P., Bechtel, B., Keramitsoglou, I., and Kiranoudis, C. T. (2018). Mapping the Spatiotemporal Dynamics of Europe's Land Surface Temperatures. *IEEE Geoscience and Remote Sensing Letters*, 15(2):202–206.

[149] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR.

[150] Srivastava, A., Valkov, L., Russell, C., Gutmann, M. U., and Sutton, C. (2017). Veegan: Reducing mode collapse in gans using implicit variational learning. *Advances in neural information processing systems*, 30.

[151] Srivastava, N., Mansimov, E., and Salakhudinov, R. (2015). Unsupervised learning of video representations using LSTMs. In *International conference on machine learning*, pages 843–852. PMLR.

[152] Su, L. and White, H. (2007). A consistent characteristic function-based test for conditional independence. *Journal of Econometrics*, 141(2):807–834.

[153] Su, L. and White, H. (2014). Testing conditional independence via empirical likelihood. *Journal of Econometrics*, 182(1):27–44.

[154] Szummer, M. and Picard, R. W. (1996). Temporal texture modeling. In *International Conference on Image Processing*, volume 3.

[155] Tansey, W., Veitch, V., Zhang, H., Rabadan, R., and Blei, D. M. (2018). The holdout randomization test: Principled and easy black box feature selection. *arXiv preprint arXiv:1811.00645*.

[156] Taylor, B. M., Davies, T. M., Rowlingson, B. S., and Diggle, P. J. (2015). Bayesian inference and data augmentation schemes for spatial, spatiotemporal and multivariate log-gaussian cox processes in R. *Journal of Statistical Software*, 63(7):1–48.

[157] Thanh-Tung, H. and Tran, T. (2020). Catastrophic forgetting and mode collapse in gans. In *2020 international joint conference on neural networks (ijcnn)*, pages 1–10. IEEE.

[158] Theis, L., Oord, A. v. d., and Bethge, M. (2015). A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*.

[159] Thomas, M. and Joy, A. T. (2006). *Elements of information theory*. Wiley-Interscience.

[160] Thorpe, M. (2018). Introduction to optimal transport. *Centre for Mathematical Sciences University of Cambridge*.

[161] Tsai, J., Lee, J. T., Wang, W., Zhang, J., Cho, H., Mamo, S., Bremer, R., Gillette, S., Kong, J., Haass, N. K., Sproesser, K., Li, L., Smalley, K. S. M., Fong, D., Zhu, Y.-L., Marimuthu, A., Nguyen, H., Lam, B., Liu, J., Cheung, I., Rice, J., Suzuki, Y., Luu, C., Settachatgul, C., Shellooe, R., Cantwell, J., Kim, S.-H., Schlessinger, J., Zhang, K.

Y. J., West, B. L., Powell, B., Habets, G., Zhang, C., Ibrahim, P. N., Hirth, P., Artis, D. R., Herlyn, M., and Bollag, G. (2008). Discovery of a selective inhibitor of oncogenic b-raf kinase with potent antimelanoma activity. *Proceedings of the National Academy of Sciences*, 105(8):3041–3046.

[162] Tulyakov, S., Liu, M.-Y., Yang, X., and Kautz, J. (2018a). Mocogan: Decomposing motion and content for video generation. In *CVPR*.

[163] Tulyakov, S., Liu, M. Y., Yang, X., and Kautz, J. (2018b). MoCoGAN: Decomposing Motion and Content for Video Generation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1526–1535.

[164] Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., and Gelly, S. (2018). Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*.

[165] Uria, B., Côté, M.-A., Gregor, K., Murray, I., and Larochelle, H. (2016). Neural autoregressive distribution estimation. *The Journal of Machine Learning Research*, 17(1):7184–7220.

[166] Uria, B., Murray, I., and Larochelle, H. (2013). Rnade: The real-valued neural autoregressive density-estimator. *arXiv preprint arXiv:1306.0186*.

[167] van den Oord, A. and Dambre, J. (2015). Locally-connected transformations for deep gmms. In *International Conference on Machine Learning (ICML): Deep learning Workshop*, pages 1–8.

[168] van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. W., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. In *ISCA workshop*.

[169] Van Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. (2016). Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR.

[170] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *NeurIPS*.

[171] Villegas, R., Yang, J., Hong, S., Lin, X., and Lee, H. (2017). Decomposing motion and content for natural video sequence prediction. *ICLR*.

[172] Vondrick, C., Pirsiavash, H., and Torralba, A. (2016). Generating videos with scene dynamics. In *NeurIPS*.

[173] Wallace, J. M. (2014). Space-time correlations in turbulent flow: A review. *Theoretical and Applied Mechanics Letters*, 4(2):022003.

[174] Wang, L., Zhang, J., Wang, Y., Lu, H., and Ruan, X. (2020a). CLIFFNet for Monocular Depth Estimation with Hierarchical Embedding Loss. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 12350 LNCS, pages 316–331. Springer Science and Business Media Deutschland GmbH.

[175] Wang, R., Kashinath, K., Mustafa, M., Albert, A., and Yu, R. (2020b). Towards Physics-informed Deep Learning for Turbulent Flow Prediction. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[176] Wang, X., Hong, Y., et al. (2018). Characteristic function based testing for conditional independence: a nonparametric regression approach. *Econometric Theory*, 34(4):815–849.

[177] Wang, X., Pan, W., Hu, W., Tian, Y., and Zhang, H. (2015). Conditional distance correlation. *Journal of the American Statistical Association*, 110(512):1726–1734.

[178] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612.

[179] Watson, D., Ho, J., Norouzi, M., and Chan, W. (2021). Learning to efficiently sample from diffusion probabilistic models. *arXiv preprint arXiv:2106.03802*.

[180] Wei, L.-Y. and Levoy, M. (2000). Fast texture synthesis using tree-structured vector quantization. In *Annual conference on Computer graphics and interactive techniques*.

[181] Weissenborn, D., Täckström, O., and Uszkoreit, J. (2020). Scaling autoregressive video models. *ICLR*.

[182] Wenliang, L. K. and Sahani, M. (2019). A neurally plausible model for online recognition and postdiction in a dynamical environment. In *NeurIPS*.

[183] Westerholt, R., Resch, B., Mocnik, F. B., and Hoffmeister, D. (2018). A statistical test on the local effects of spatially structured variance. *International Journal of Geographical Information Science*, 32(3):571–600.

[184] Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

[185] Xu, T. and Acciaio, B. (2021). Conditional cot-gan for video prediction with kernel smoothing. *arXiv preprint arXiv:2106.05658*.

[186] Xu, T., Wenliang, L. K., Munn, M., and Acciaio, B. (2020a). COT-GAN: Generating Sequential Data via Causal Optimal Transport. In *NeurIPS*.

[187] Xu, T., Wenliang, L. K., Munn, M., and Acciaio, B. (2020b). COT-GAN: Generating sequential data via causal optimal transport. In *Advances in Neural Information Processing Systems*, volume 2020-Decem. Neural information processing systems foundation.

[188] Yan, B., Mai, G., Janowicz, K., and Gao, S. (2017). From ITDL to Place2Vec – Reasoning About Place Type Similarity and Relatedness by Learning Embeddings From Augmented Spatial Contexts. In *SIGSPATIAL: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*.

[189] Yin, Y., Liu, Z., Zhang, Y., Wang, S., Shah, R. R., and Zimmermann, R. (2019). GPS2Vec: Towards generating worldwide GPS embeddings. In *SIGSPATIAL: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, pages 416–419, New York, NY, USA. Association for Computing Machinery.

[190] Yoon, J., Jarrett, D., and van der Schaar, M. (2019). Time-series generative adversarial networks. In *NeurIPS*.

[191] Yuan, Y., Cave, M., and Zhang, C. (2018). Using Local Moran's I to identify contamination hotspots of rare earth elements in urban soils of London. *Applied Geochemistry*, 88:167–178.

[192] Zammit-Mangion, A., Ng, T. L. J., Vu, Q., and Filippone, M. (2019). Deep Compositional Spatial Models.

[193] Zhang, J. and Zhao, X. (2021). Spatiotemporal wind field prediction based on physics-informed deep learning and LIDAR measurements. *Applied Energy*, 288:116641.

[194] Zhang, K., Peters, J., Janzing, D., and Schölkopf, B. (2011). Kernel-based conditional independence test and application in causal discovery. In *27th Conference on Uncertainty in Artificial Intelligence (UAI 2011)*, pages 804–813. AUAI Press.

[195] Zhang, M., Hayes, P., Bird, T., Habib, R., and Barber, D. (2020a). Spread divergence. In *International Conference on Machine Learning*, pages 11106–11116. PMLR.

[196] Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595.

[197] Zhang, X. L., Begleiter, H., Porjesz, B., Wang, W., and Litke, A. (1995). Event related potentials during object recognition tasks. *Brain research bulletin*, 38(6):531–538.

[198] Zhang, Y., Fu, Y., Wang, P., Li, X., and Zheng, Y. (2019). Unifying inter-region autocorrelation and intra-region structures for spatial embedding via collective adversarial learning. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[199] Zhang, Y., Li, Y., Zhou, X., Kong, X., and Luo, J. (2020b). Curb-GAN: Conditional Urban Traffic Estimation through Spatio-Temporal Generative Adversarial Networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume 20, pages 842–852, New York, NY, USA. Association for Computing Machinery.

[200] Zhou, S., Gordon, M., Krishna, R., Narcomey, A., Fei-Fei, L. F., and Bernstein, M. (2019). Hype: A benchmark for human eye perceptual evaluation of generative models. In *NeurIPS*.

# Appendix A

# COT-GAN

## A.1 Specifics on regularized Causal Optimal Transport

### A.1.1 The MMD limiting case.

In the limit $\varepsilon \to \infty$, Genevay et al. [62] showed that $\mathcal{W}_{c,\varepsilon}(\mu,\nu) \to \text{MMD}_{-c}(\mu,\nu)$ under the kernel defined by $-c(x,y)$. Here we want to point out an interesting fact about the limiting behavior of the mixed Sinkhorn divergence.

**Remark A.1.1.** Given distributions of mini-batches $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ formed by samples from $\mu$ and $\nu$, respectively, in the limit $\varepsilon \to \infty$, the Sinkhorn divergence $\widehat{\mathcal{W}}_{c,\varepsilon}(\hat{\mathbf{x}},\hat{\mathbf{y}})$ converges to a biased estimator of $\text{MMD}_{-c}(\mu,\nu)$; given additional $\hat{\mathbf{x}}'$ and $\hat{\mathbf{y}}'$ from $\mu$ and $\nu$, respectively, the mixed Sinkhorn divergence $\widehat{\mathcal{W}}^{\text{mix}}_{c,\varepsilon}(\hat{\mathbf{x}},\hat{\mathbf{x}}',\hat{\mathbf{y}},\hat{\mathbf{y}}')$ converges to an unbiased estimator of $\text{MMD}_{-c}(\mu,\nu)$.

*Proof.* The first part of the statement relies on the fact that $\text{MMD}_{-c}(\hat{\mathbf{x}},\hat{\mathbf{y}})$ is a biased estimator of $\text{MMD}_{-c}(\mu,\nu)$. Indeed, we have

$$\widehat{\mathcal{W}}_{c,\varepsilon}(\hat{\mathbf{x}},\hat{\mathbf{y}}) \xrightarrow{\varepsilon\to\infty} \text{MMD}_{-c}(\hat{\mathbf{x}},\hat{\mathbf{y}}) = -\frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}[c(x^i,x^j)+c(y^i,y^j)-2c(x^i,y^j)].$$

Now note that

$$\begin{aligned}\frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\mathbb{E}[c(x^i,x^j)] &= \frac{1}{m^2}\left[\sum_{i=1}^{m}\mathbb{E}_{\mu}[c(x^i,x^i)]+\sum_{i\neq j}\mathbb{E}_{\mu\otimes\mu}[c(x^i,x^j)]\right]\\ &= \frac{m-1}{m}\mathbb{E}_{\mu\otimes\mu}[c(x,x')],\end{aligned}$$

where we have used the fact that $c(x^i, x^i) = 0$. A similar result holds for the sum over $c(y^i, y^j)$. On the other hand, $\frac{1}{m^2}\sum_{ij}\mathbb{E}[c(x^i, y^j)] = \mathbb{E}_{\mu\otimes\nu}[c(x,y)]$. Therefore

$$\begin{aligned}\mathbb{E}\,\mathrm{MMD}_{-c}(\hat{\mathbf{x}},\hat{\mathbf{y}}) &= -\frac{m-1}{m}[\mathbb{E}_{\mu\otimes\mu}[c(x,x')] + \mathbb{E}_{\nu\otimes\nu}[c(y,y')]] + 2\mathbb{E}_{\mu\otimes\nu}[c(x,y)] \\ &\neq \mathrm{MMD}_{-c}(\mu,\nu),\end{aligned}$$

which completes the proof of the first part of the statement.

For the second part, note that $\mathcal{W}_{c,\varepsilon}(\mu,\nu) \to \mathbb{E}_{\mu\otimes\mu}[c(x,x')]$ as $\varepsilon \to \infty$ [62, Theorem 1], thus

$$\begin{aligned}\widehat{\mathcal{W}}^{\mathrm{mix}}_{c,\varepsilon}(\hat{\mathbf{x}},\hat{\mathbf{x}}',\hat{\mathbf{y}},\hat{\mathbf{y}}') &\to \mathbb{E}_{\hat{\mathbf{x}}\otimes\hat{\mathbf{y}}}[c(x,y)] + \mathbb{E}_{\hat{\mathbf{x}}'\otimes\hat{\mathbf{y}}'}[c(x',y')] - \mathbb{E}_{\hat{\mathbf{x}}\otimes\hat{\mathbf{x}}'}[c(x,x')] - \mathbb{E}_{\hat{\mathbf{y}}\otimes\hat{\mathbf{y}}'}[c(y,y')] \\ &= \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}[c(x^i,y^i) + c(x'^i,y'^i) - c(x^i,x'^i) - c(y^i,y'^i)].\end{aligned}$$

The RHS is an unbiased estimator of MMD, since its expectation is

$$\mathbb{E}_{\mu\otimes\nu}[c(x,y)] + \mathbb{E}_{\mu\otimes\nu}[c(x',y')] - \mathbb{E}_{\mu\otimes\mu}[c(x,x')] - \mathbb{E}_{\nu\otimes\nu}[c(y,y')] = \mathrm{MMD}_{-c}(\mu,\nu).$$

□

The mixed divergence may still be a biased estimate of the true Sinkhorn divergence. However, in the experiment of Example 3.2.4 we note that the minimum is reached for the parameter $\theta$ close to the real one (Figure 3.1, bottom).

## A.2 Experimental details

### A.2.1 Low dimensional time series

Here we describe details of the experiments in Section 3.4.1.

**Autoregressive process.** The generative process to obtain data $\mathbf{x}_t$ for the autoregressive process is

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \boldsymbol{\zeta}_t, \quad \boldsymbol{\zeta}_t \overset{\text{i.i.d}}{\sim} \mathcal{N}(0,\boldsymbol{\Sigma}), \quad \boldsymbol{\Sigma} = 0.5\mathbf{I} + 0.5,$$

where $\mathbf{A}$ is diagonal with ten values evenly spaced between 0.1 and 0.9. We initialize $\mathbf{x}_0$ from a 10-dimensional standard normal, and ignore the data in the first 10 time steps so that the data sequence begins with a more or less stationary distribution. We use $\lambda = 0.1$ and $\varepsilon = 10.0$ for this experiment.

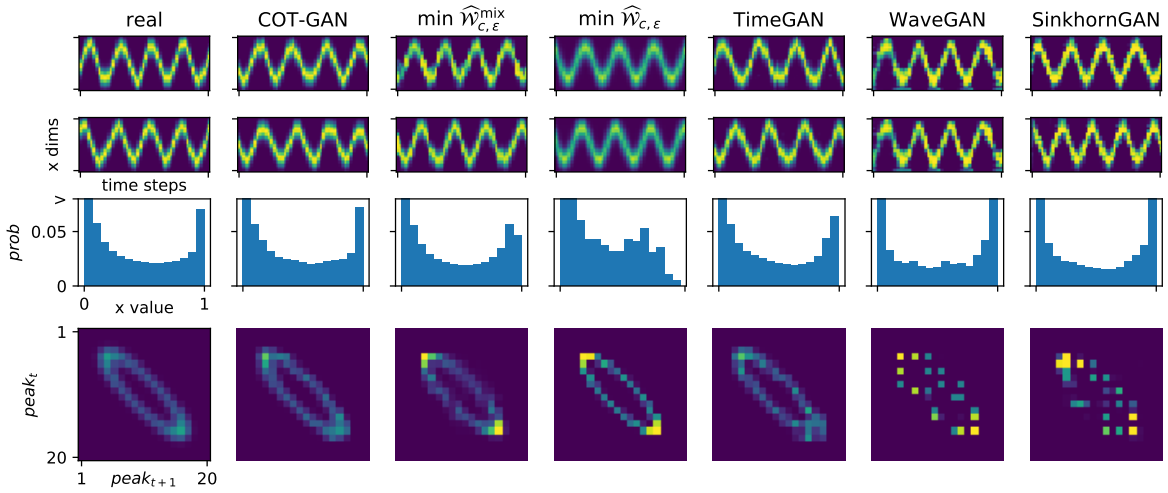



Fig. A.1 1-D noisy oscillation. Same distributions as in 3.7 are shown.

**Noisy oscillation.** This dataset comprises paths simulated from a noisy, nonlinear dynamical system. Each path is represented as a sequence of $d$-dimensional arrays, $T$ time steps long, and can be displayed as a $d \times T$-pixel image for visualization. At each discrete time step $t \in \{1, \ldots, T\}$, data at time $t$, given by $\mathbf{x}_t \in [0,1]^d$, is determined by the position of a "particle" following noisy, nonlinear dynamics. When shown as an image, each sample path appears visually as a "bump" travelling rightward, moving up and down in a zig-zag pattern as shown in Figure A.1 (top left).

More precisely, the state of the particle at time $t$ is described by its position and velocity $\mathbf{s}_t = (s_{t,1}, s_{t,2}) \in \mathbb{R}^2$, and evolves according to

$$\mathbf{s}_t = \mathbf{f}(\mathbf{s}_{t-1}) + \boldsymbol{\zeta}_t, \quad \boldsymbol{\zeta}_t = \mathcal{N}(0, 0.1\mathbf{I}),$$
$$\mathbf{f}(\mathbf{s}_{t-1}) = c_t \mathbf{A}\mathbf{s}_{t-1}; \quad c_t = \frac{1}{\|\mathbf{s}_{t-1}\|_2 \exp(-4(\|\mathbf{s}_{t-1}\|_2 - 0.3) + 1)},$$

where $\mathbf{A} \in \mathbb{R}^{2\times 2}$ is a rotation matrix, and $\mathbf{s}_0$ is uniformly distributed on the unit circle.

We take $T = 48$ and $d = 20$ so that $\mathbf{x}_t$ is a vector of evaluations of a Gaussian function at 20 evenly spaced locations, and the peak of the Gaussian function follows the position of the particle $s_{t,1}$ for each $t$:

$$x_{t,i} = \exp\left[-\frac{(\text{loc}(i) - s_{t,1})^2}{2 \times 0.3^2}\right],$$

where $\text{loc} : \{1, \ldots, d\} \to \mathbb{R}$ maps pixel indices to a grid of evenly spaced points in the space of particle position. Thus, $\mathbf{x}_t$, the observation at time $t$, contains information about $s_{t,1}$ but not $s_{t,2}$. A similar data generating process was used in [182], inspired by **?** ].

We compare the marginal distribution of the pixel values $x_{t,i}$ and joint distribution of the bump location ($\arg\max_i x_{t,i}$) between adjacent time steps. See Figure A.1.

**Electroencephalography.** For COT-GAN, we train three variants corresponding to $\lambda$ being one of $\{1.0, 0.1, 0.01\}$, and $\varepsilon = 100.0$ for all OT-based methods.

**Model and training parameters.** The dimensionality of the latent state is 10 at each time step, and there is also a 10-dimensional time-invariant latent state. The generator common to COT-GAN, direct minimization and SinkhornGAN comprise a 1-layer (synthetic) or 2-layer (EEG) LSTM networks, whose output at each time step is passed through two layers of fully connected ReLU networks. We used Adam for updating $\theta$ and $\varphi$, with learning rate 0.001. Batch size is 32 for all methods except for direct minimization of the mixed and original Sinkhorn divergence which is trained with batch size 64. These hyperparameters do not substantially affect the results.

The same discriminator architecture is used for both $h$ and $M$ in COT-GAN and the discriminator of the SinkhornGAN. This network has two layers of 1-D causal CNN with stride 1, filter length 5. Each layer has 32 (synthetic data) or 64 neurons (EEG) at each time step. The activation is ReLU except at the output which is linear for autoregressive process, sigmoid for noisy oscillation, and tanh for EEG.

For COT-GAN, $\lambda = 10.0$ and $\varepsilon = 10$ for synthetic datasets, and $\lambda \in \{0.01, 0.1, 1.0\}$ and $\varepsilon = 100.0$ for EEG. The choice of $\varepsilon$ is made based on how fast it converges to a particular threshold of the transport plan, and each iteration takes around 1 second on a 2.6GHz Xeon CPU.

### A.2.2 Videos datasets

#### Sprite animations

**Data pre-processing.** The sprite sheets can be created and downloaded from [1]. The data can be generated with various feature options for clothing, hairstyle and skin color, etc. Combining all feature options gives us 6352 characters in total. Each character performs spellcast, walk, slash, shoot and hurt movements from different directions, making up to a total number of 21 actions. As the number of frames $T$ ranges from 6 to 13, we pad all actions to have the same length $T = 13$ by repeating previous movements in shorter

[1] Original dataset is available at gaurav.munjal.us/Universal-LPC-Spritesheet-Character-Generator/ and github.com/jrconway3/Universal-LPC-spritesheet. To facilitate the use of large dataset in TensorFlow, we pre-shuffled all data used and wrote into tfrecord files. Links for download can be found on the Github repository.

sequences. We then crop the characters from sheets to be in the center of each frame, which gives a dimension of $64 \times 64 \times 4$ for each frame. We decide to drop the 4th color channel (alpha channel) to be consistent with the input setting of baseline models. Finally, the resulting dataset has 6352 data points consisting of sequences with 13 frames of dimensions $64 \times 64 \times 3$.

### The Weizmann Action database

**Data pre-processing.** The videos in this dataset consists of clips that have lengths from 2 to 7 seconds. Each second of the original videos contains 25 frames, each of which has dimension 144x180x3. To avoid the absence of objects at the beginning of the videos and to ensure an entire evolution of motions in each sequence, we skip the first 5 frames, then skip every 2 frames and collect 16 frames in a whole sequence as a result. Due to limited access to hardware, we also downscale each frame to $64 \times 64 \times 3$. The training set used contains 89 data points with dimensions $16 \times 64 \times 64 \times 3$.

**GAN architectures.** We detail the GAN architectures used in the experiment of the Weizmann Action database in Table C.1 and Table C.2. A latent variable $z$ of shape $5 \times 5$ per time step is sampled from a multivariate standard normal distribution and is then passed to a 2-layer LSTM to generate time-dependent features, followed by 4-layer deconvolutional neural network (DCONV) to map the features to frames. In order to connect two different types of networks, we map the features using a feedforward (dense) layer and reshape them to the desired shape for DCNN. In Table C.1 and C.2, the DCONV layers have N filter size, K kernel size, S strides and P padding option. We adopted batch-normalisation layers and the LeakyReLU activation function. We have two networks to parameterize the process $h$ and $M$ as discriminator share the same structure, shown in Table C.2.

We use a fixed length $T = 16$ of LSTM. The state size in the last LSTM layer corresponds to the dimensions of $h_t$ and $M_t$, i.e., $j$ in (3.2.13). We also applied exponential decay to learning rate by $\eta_t = \eta_0 r^{s/c}$ where $\eta_0$ is the initial learning rate, $r$ is decay rate, $s$ is the current number of training steps and $c$ is the decaying frequency. In our experiments, we set the initial learning rate to be 0.001, decay rate 0.98, and decaying frequency 500. The batch size $m$ and time steps $T$ used are both 16. We have $\lambda = 0.01$, $\varepsilon = 6.0$ and the Sinkhorn $L = 100$ in this experiment. We train COT-GAN on a single NVIDIA Tesla P100 GPU for 3 or 4 days. Each iteration takes roughly 1.5 seconds.

Table A.1 Generator architecture.

| Generator | Configuration |
|---|---|
| Input | $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ |
| 0 | LSTM(state size = 128), BN |
| 1 | LSTM(state size = 256), BN |
| 2 | Dense(8*8*512), BN, LeakyReLU |
| 3 | reshape to 4D array of shape (m, 8, 8, 512) as input for DCONV |
| 4 | DCONV(N512, K5, S1, P=SAME), BN, LeakyReLU |
| 5 | DCONV(N256, K5, S2, P=SAME), BN, LeakyReLU |
| 6 | DCONV(N128, K5, S2, P=SAME), BN, LeakyReLU |
| 7 | DCONV(N3, K5, S2, P=SAME) |

Table A.2 Discriminator architecture.

| Discriminator | Configuration |
|---|---|
| Input | 64x64x3 |
| 0 | CONV(N128, K5, S2, P=SAME), BN, LeakyReLU |
| 1 | CONV(N256, K5, S2, P=SAME), BN, LeakyReLU |
| 2 | CONV(N512, K5, S2, P=SAME), BN, LeakyReLU |
| 3 | reshape to 3D array of shape (m, T, -1) as input for LSTM |
| 4 | LSTM(state size = 512), BN |
| 5 | LSTM(state size = 128) |

## A.3 Sprites and human action results without cherry-picking

In this section we show random samples of Sprites and human actions generated by COT-GAN, mixed Sinkhorn minimization, and MoCoGAN without cherry-picking. The background was static for both experiments. In the Sprites experiments (see Figure A.2), the samples from mixed Sinkhorn minimization and COT-GAN are both of good quality, whereas those from MoCoGAN only capture a rough pattern in the frames and fail to show a smooth evolution of motions.

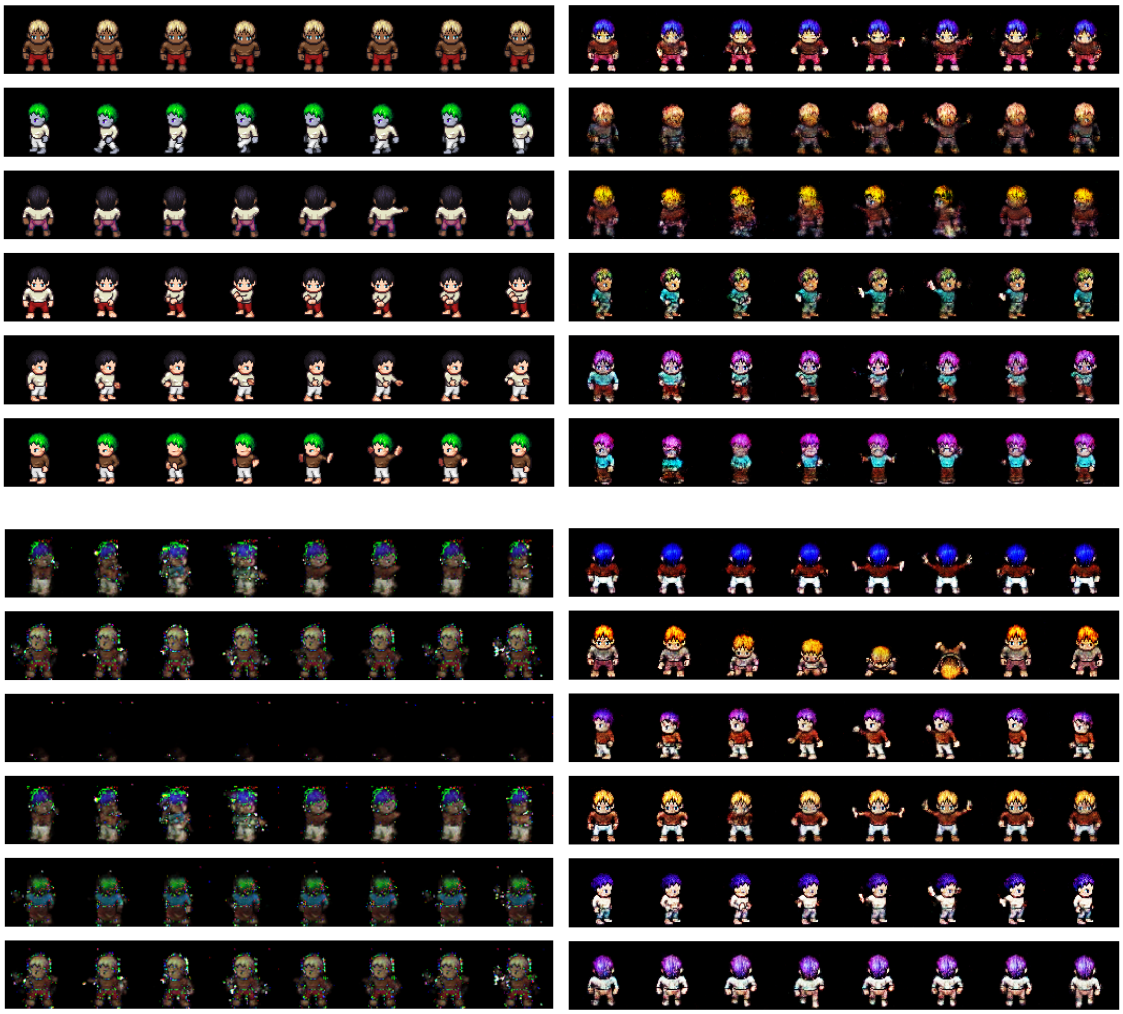


Fig. A.2 Random samples with no cherry picking from models trained on animated Sprites. Top row: real sequences on the left and mixed Sinkhorn minimization on the right; bottom row: MoCoGAN on the left and COT-GAN on the right.

In Figure A.3, we show a comparison of real and generated samples for human action sequences. Noticeable artifacts of COT-GAN and mixed Sinkhorn minimization results

include blurriness and even disappearance of the person in a sequence, which normally happens when the clothing of the person has a similar color as the background. MoCoGAN also suffers from this issue and, visually, there appears to be some degree of mode collapse. We used generators of similar capacity across all models and trained COT-GAN, mixed Sinkhorn minimization and MoCoGAN for 65000 iterations.

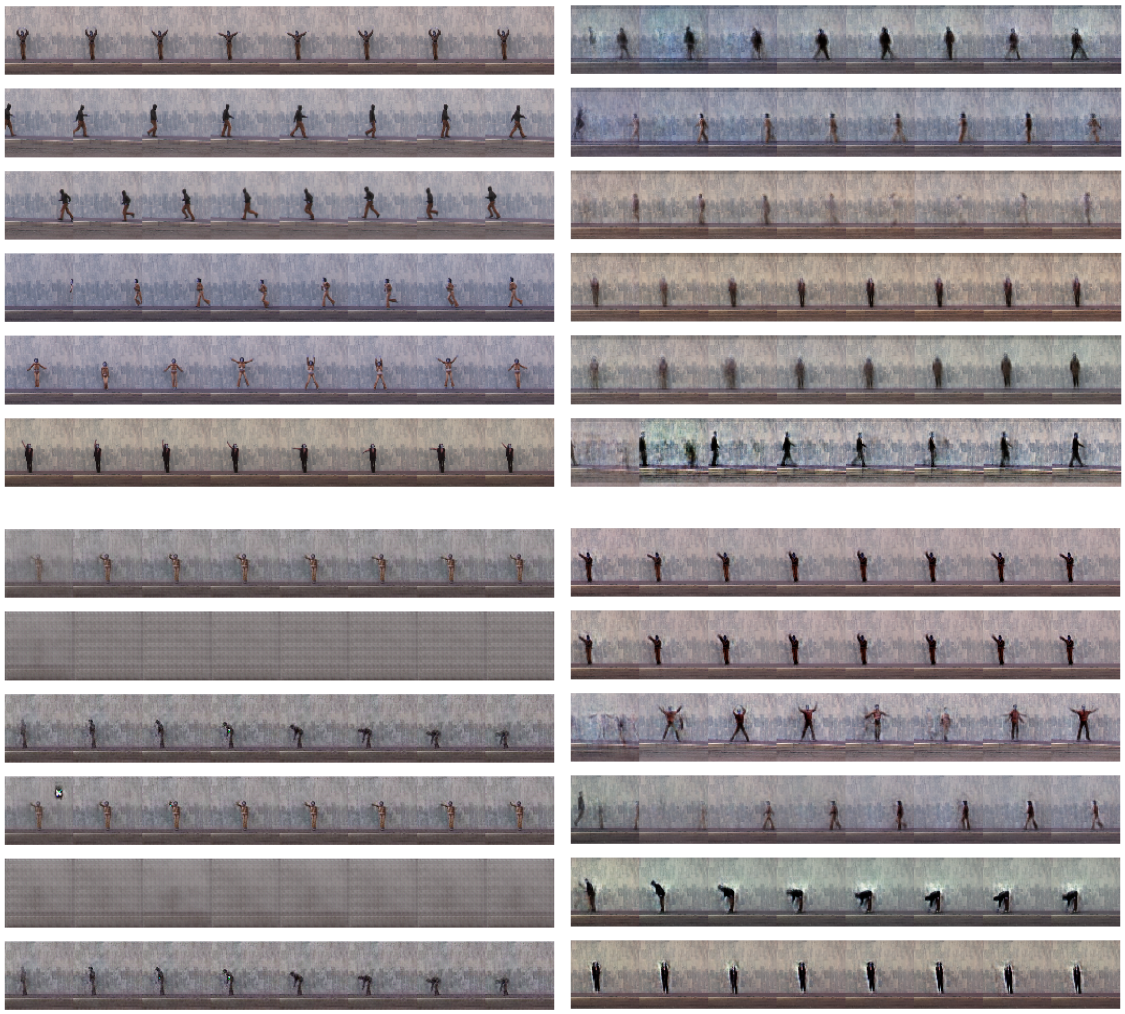


Fig. A.3 Random samples with no cherry picking from models trained on human actions. Top row: real sequences on the left and mixed Sinkhorn minimization on the right; bottom row: MoCoGAN on the left and COT-GAN on the right.

# Appendix B

# KCCOT-GAN

## B.1 Experiment details

### B.1.1 Network architectures and training details

All experiments on the three datasets share the same GAN architectures. The generator is split into an encoder and a decoder, supported by convolutional LSTM (convLSTM). The encoder learns both the spatial and temporal features of the input sequences, whereas the decoder predicts the future evolution conditioned on the learned features and a latent variable.

The features from the last encoding layer has a shape of $4 \times 4$ (height $\times$ width) per time step. A latent variable $z$ is sampled from a multivariate standard normal distribution with the same shape as the features (same number of channels too depending on the model size). We then concatenate the features, input sequence, and latent variables over the channel dimension as input for the decoder. The encoder and decoder structures are detailed in Table C.1. As the discriminator, the process $\mathbf{h}$ and $\mathbf{M}$ are parameterized with two separate networks that share the same structure, shown in Table C.2. In all tables, we use DCONV to represent a de-convolutional (convolutional transpose) layer. The layers may have N filter size, K kernel size, S strides and P padding option. We adopt both batch-normalization(BN) and layer-normalization(LN), and the LeakyReLU activation function. All hyperparameter setting are the same for all three datasets except that the filter size is halved for the Moving MNIST dataset.

During training, we apply exponential decay to the learning rate by $\eta_t = \eta_0 r^{s/c}$ where $\eta_0$ is the initial learning rate, $r$ is decay rate, $s$ is the current number of training steps and $c$ is the decaying frequency. The bandwidth parameter $h$ are also annealed from 1.5 to 0.1 in a similar manner. In all experiments, the initial learning rate is 0.0005, decay rate 0.985, decaying frequency 10000, and batch size $m = 8$. The settings of hyper-parameters in the

Table B.1 Encoder and decoder architecture.

| Encoder Configuration | |
|---|---|
| Input | $x_{1:T}$ with shape $T \times 64 \times 64 \times 3$ |
| 1 | convLSTM2D(N32, K6, S2, P=SAME), LN |
| 2 | convLSTM2D(N64, K6, S2, P=SAME), LN |
| 3 | convLSTM2D(N128, K5, S2, P=SAME), LN |
| 4 | convLSTM2D(N256, K5, S2, P=SAME), LN |
| 5 | output features $e_{1:T}$ with shape $T \times 4 \times 4 \times 256$ |
| **Decoder Configuration** | |
| Input | $z_{k:T-1}$, $e_{k:T-1}$, $x_{k:T-1}$ |
| 1 | DCONV(N256, K2, S2, P=SAME), LN |
| 2 | convLSTM2D(N128, K4, S1, P=SAME), LN |
| 3 | DCONV(N128, K4, S2, P=SAME), LN |
| 4 | convLSTM2D(N64, K6, S1, P=SAME), LN |
| 5 | DCONV(N64, K6, S2, P=SAME), LN |
| 6 | convLSTM2D(N32, K6, S1, P=SAME), LN |
| 4 | DCONV(N16, K6, S1, P=SAME), LN |
| 5 | convLSTM2D(N8, K8, S1, P=SAME), LN |
| 7 | DCONV(N3, K8, S1, P=SAME), Sigmoid |

Table B.2 Discriminator architecture.

| Discriminator | Configuration |
|---|---|
| Input | 64x64x3 |
| 0 | CONV(N32, K5, S2, P=SAME), BN |
| 1 | CONV(N64, K5, S2, P=SAME), BN |
| 2 | CONV(N128, K5, S2, P=SAME), BN |
| 3 | reshape 3D array for LSTM |
| 4 | LSTM(state size = 128), LN |
| 5 | LSTM(state size = 64), LN |
| 6 | LSTM(state size = 32), LN |

Sinkhorn algorithm are also shared across the three datasets with $\lambda = 1.0$, $\varepsilon = 0.8$ and the Sinkhorn iterations $L = 100$. We train KCCOT-GAN and CCOT-GAN on a single NVIDIA GTX 1080 Ti GPU. Each iteration takes roughly 3.5 seconds. Each experiment is run for around 100000 iterations.

### B.1.2 KKCOT-GAN results on Moving MNIST

Predictions for the Moving MNIST test set are presented in Figure B.1.

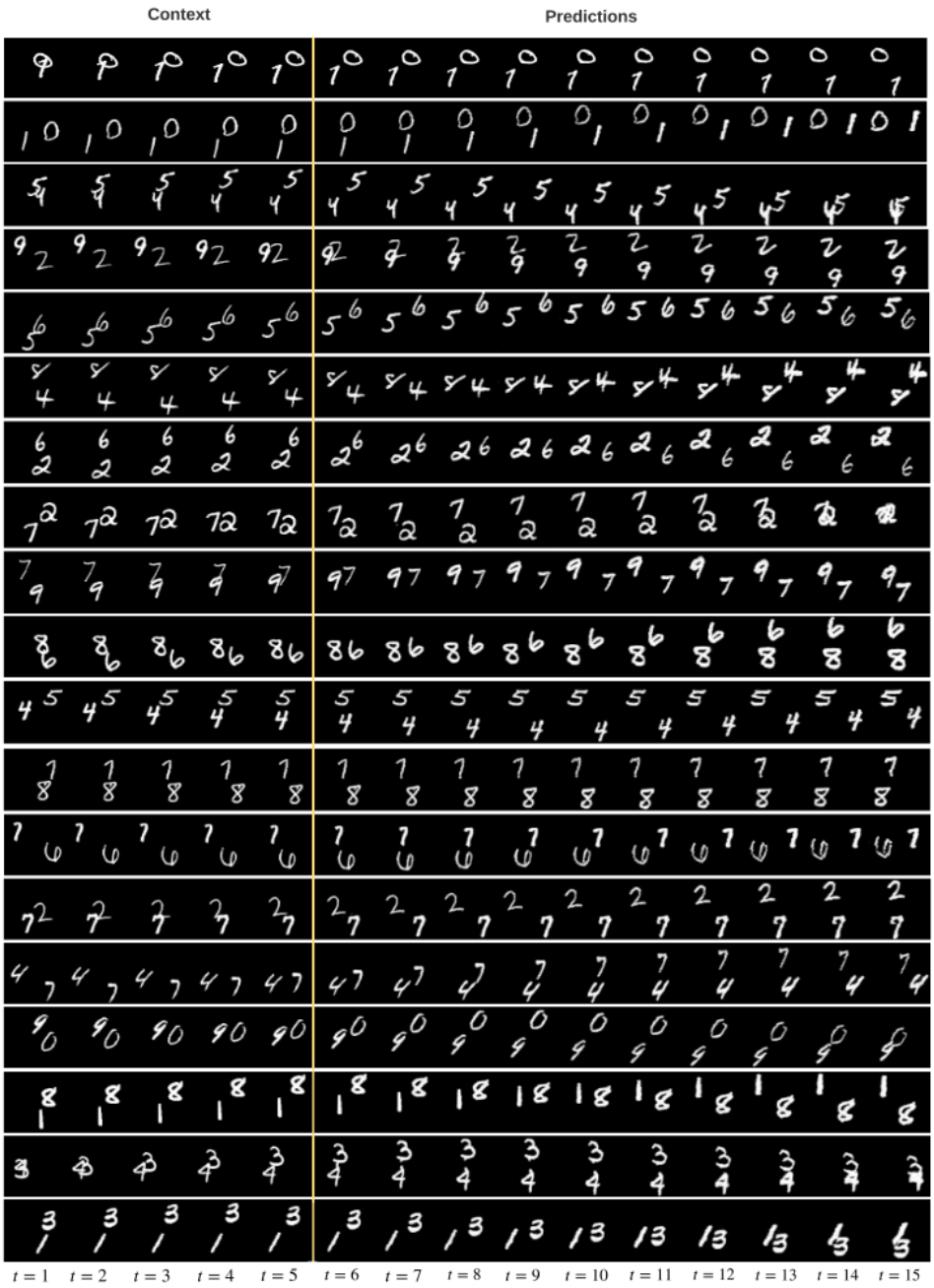



Fig. B.1 Moving MNIST results on test set. The first 5 frames are context sequence and last 10 frames are predictions from KCCOT-GAN, separated by the yellow vertical line.

# Appendix C

# SPATE-GAN

## C.1 Training details

We used a smaller size of model with the same network architectures as COT-GAN to train all three datasets. The architectures for generator and discriminator are given in Tables C.1 and C.2.

Hyperparameter settings are as follows: the Sinkhorn regularizer $\varepsilon = 0.8$, Sinkhorn iteration $L = 100$, the lengthscale $l = 20$ and martingale penalty $\lambda = 1.5$. We used Adam optimizer with learning rate 0.0001, $\beta_1 = 0.5$ and $\beta_2 = 0.9$. All models are trained for $60,000$ iterations.

Table C.1 Generator architecture.

| Generator | Configuration |
|---|---|
| Input | $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ |
| 0 | LSTM(state size = 64), BN |
| 1 | LSTM(state size = 128), BN |
| 2 | Dense(8*8*256), BN, LeakyReLU |
| 3 | reshape to 4D array of shape (m, 8, 8, 256) |
| 4 | DCONV(N256, K5, S1, P=SAME), BN, LeakyReLU |
| 5 | DCONV(N128, K5, S2, P=SAME), BN, LeakyReLU |
| 6 | DCONV(N64, K5, S2, P=SAME), BN, LeakyReLU |
| 7 | DCONV(N1, K5, S2, P=SAME) |

Table C.2 Discriminator architecture.

| Discriminator | Configuration |
|---|---|
| Input | |
| 0 | CONV(N64, K5, S2, P=SAME), BN, LeakyReLU |
| 1 | CONV(N128, K5, S2, P=SAME), BN, LeakyReLU |
| 2 | CONV(N256, K5, S2, P=SAME), BN, LeakyReLU |
| 3 | reshape to 3D array of shape (m, T, -1) |
| 4 | LSTM(state size = 256), BN |
| 5 | LSTM(state size = 64) |

## C.2 Evaluation metrics

To compute our three metrics, let us first assume that we have a set of real data samples ($\mathcal{P}$) and synthetic data samples ($\mathcal{S}$). EMD is defined as:

$$EMD(\mathcal{P},\mathcal{S}) = \min_{\phi:\mathcal{P}\to\mathcal{S}} \sum_{p\in\mathcal{P}} \|p-\phi(p)\| \tag{C.2.1}$$

where $\phi : \mathcal{P} \to \mathcal{S}$ is a bijection. MMD is defined as:

$$\begin{aligned}\widehat{MMD}^2(\mathcal{P},\mathcal{S}) = \frac{1}{n(n-1)}\sum k(p,p) + \\ \frac{1}{n(n-1)}\sum k(s,s) - \frac{2}{n^2}\sum k(p,s)\end{aligned} \tag{C.2.2}$$

where $k$ denotes a positive-definite kernel (e.g. RBF kernel) and $n$ is the number of (real or synthetic) samples.

Lastly, to compute the KNN score, we first split our real and synthetic samples $\mathcal{P}$ and $\mathcal{S}$ into training and test datasets $\mathcal{D}_{tr}$ and $\mathcal{D}_{te}$ so that $\mathcal{D} = \mathcal{D}_{tr} \cup \mathcal{D}_{te}$. We train the KNN classifier $f : \mathcal{X}_{tr} \to [0,1]$ using training data. The accuracy of the trained classifier is then obtained using test samples $\mathcal{D}_{te}$ and given as:

$$\hat{t} = \frac{1}{n_{te}} \sum_{(z_i,l_i)\in\mathcal{D}_{te}} \mathbb{I}\left[\left(f(z_i) > \frac{1}{2}\right) = l_i\right] \tag{C.2.3}$$

where $f(z_i)$ estimates the conditional probability distribution $p(l = 1|z_i)$. A classifier accuracy approaching random chance (50%) indicates better synthetic data. As suggested by Lopez-Paz and Oquab [107], we use a 1-NN classifier to obtain the score.

## C.3 More figures

In this section, we provide more results in larger figures for visual comparisons.

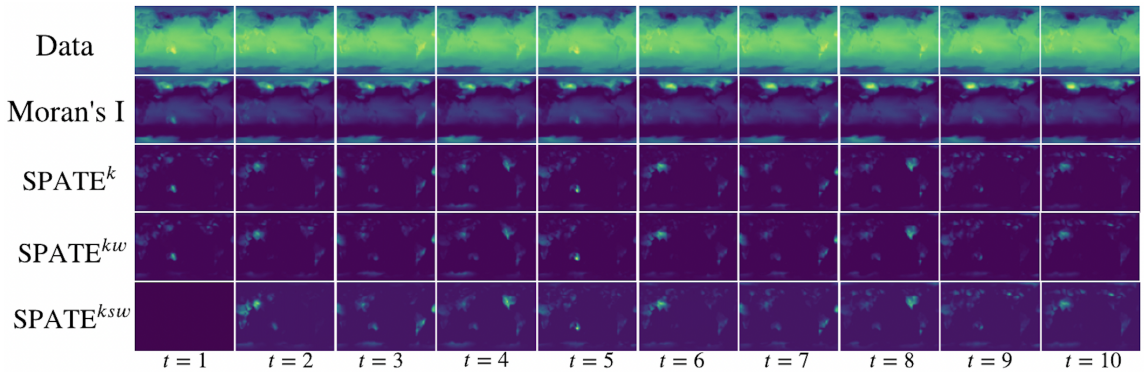



Fig. C.1 Larger version of Figure 2 for the purpose of visual comparison.

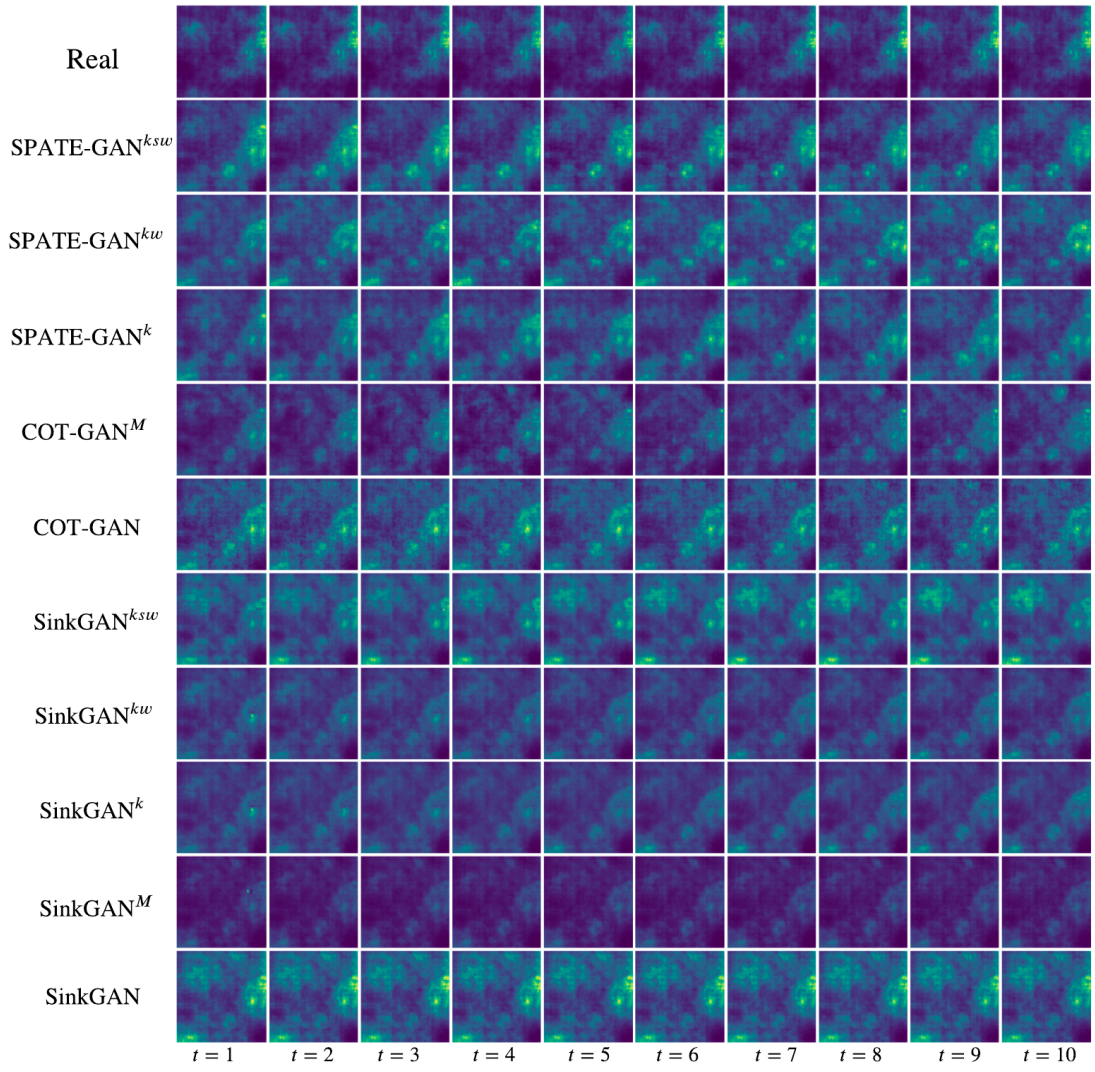



Fig. C.2 More selected samples for log-Gaussian Cox process (LGCP) dataset.

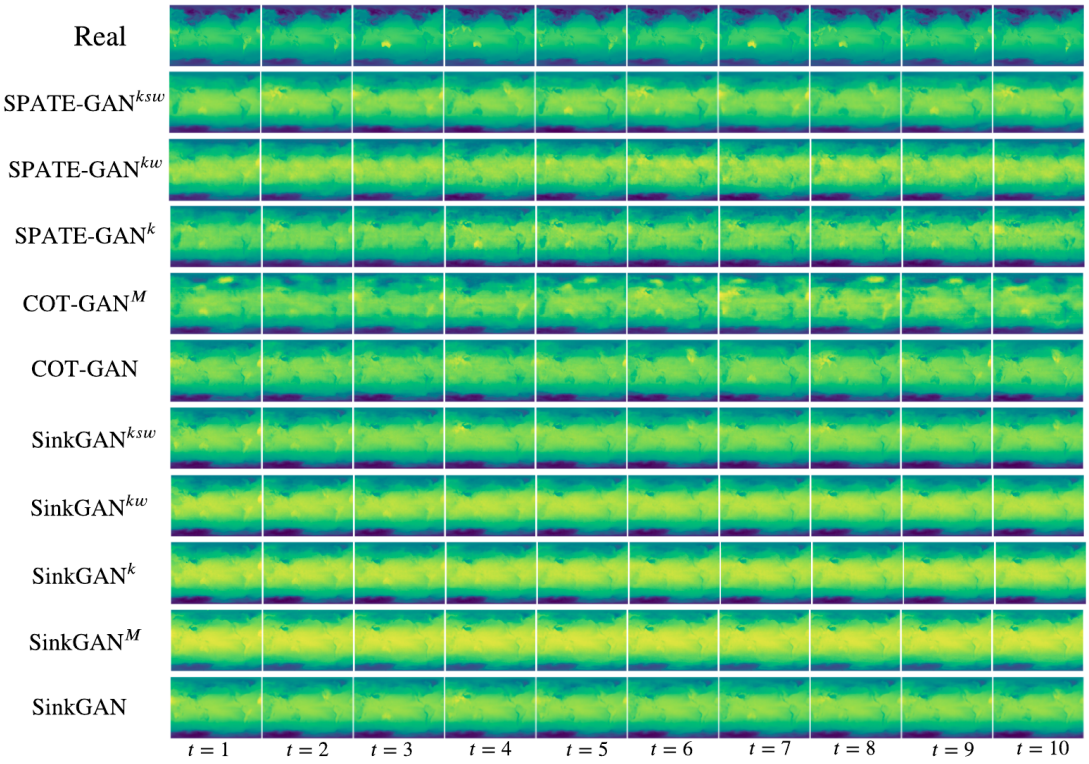



Fig. C.3 More selected samples for extreme weather (EW) dataset.

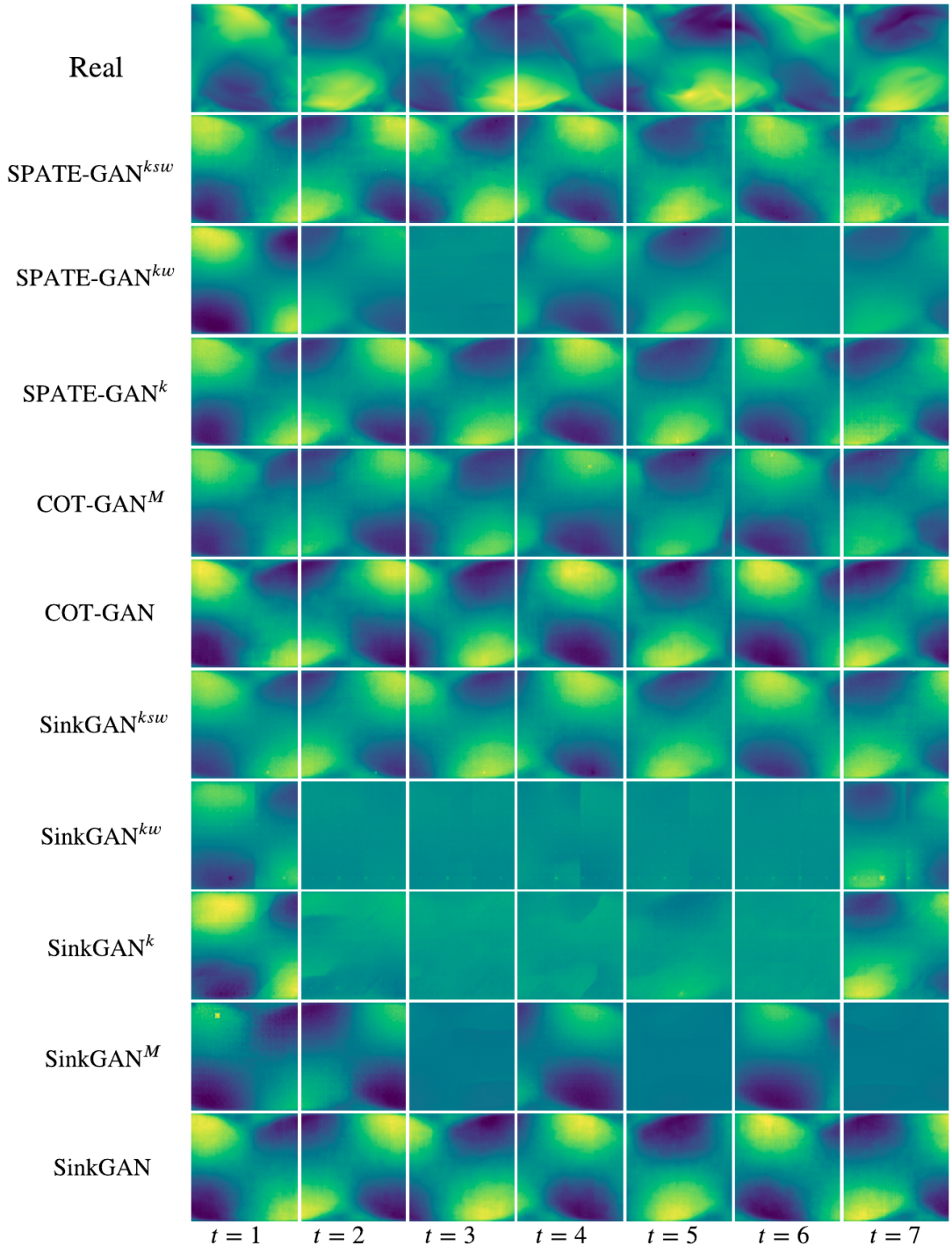



Fig. C.4 More selected samples for turbulent flow (TF) dataset.

# Appendix D

# Double GANs for Conditional Independence Testing

## D.1 Proofs

We provide the proofs of Proposition 2, Theorems 6.4.2, 6.4.3, and 6.4.4. We omit the proof of Theorem 6.4.1, since it is similar to that of Theorem 6.4.2. We note that Theorems 6.4.1-6.4.4 are established under our choice of the function classes $\mathbb{H}_1$ and $\mathbb{H}_2$, which are set to the classes of neural networks with a single-hidden layer, finitely many hidden nodes, and the sigmoid activation function, as used in our implementation. Meanwhile, our results can be extended to more general choices of the function classes.

### D.1.1 Proof of Proposition 2

Note that the total variation distance is bounded by 1. Suppose $\mathrm{E}d_{\mathrm{TV}}(\widetilde{P}_{\boldsymbol{X}|\boldsymbol{Z}}, P_{\boldsymbol{X}|\boldsymbol{Z}}) = o(1)$. Then we have $d_{\mathrm{TV}}(\widetilde{P}_{\boldsymbol{X}|\boldsymbol{Z}}, P_{\boldsymbol{X}|\boldsymbol{Z}}) = o_p(1)$. By the dominated convergence theorem, we have $\mathrm{E}d^2_{\mathrm{TV}}(\widetilde{P}_{\boldsymbol{X}|\boldsymbol{Z}}, P_{\boldsymbol{X}|\boldsymbol{Z}}) = o(1)$.

By Theorem 1.2 of [43], we have $d_{\mathrm{TV}}(\widetilde{P}_{\boldsymbol{X}|\boldsymbol{Z}}, P_{\boldsymbol{X}|\boldsymbol{Z}})$ is proportional to

$$\min\left[1, \sigma_0^{-1}\sqrt{\sum_{i=1}^{n}\left\{Z_i^\top(\widehat{\beta}-\beta_0)\right\}^2}\right].$$

It follows that

$$\frac{1}{\sigma_0}\mathrm{E}\sum_{i=1}^{n}\{Z_i^\top(\widehat{\beta}-\beta_0)\}^2 = o(1).$$

Applying Theorem 1.2 of [43] again, we obtain that $d_{\mathrm{TV}}(\widetilde{P}_{X|Z=Z_i}, P_{X|Z=Z_i})$ is proportional to

$$\min\left\{1, \sigma_0^{-1}|Z_i^\top(\widehat{\beta}-\beta_0)|\right\}.$$

Therefore, we obtain that,

$$\sum_{i=1}^n \mathrm{E} d_{\mathrm{TV}}^2\left(\widetilde{P}_{X|Z=Z_i}, P_{X|Z=Z_i}\right) = o(1).$$

Since the data is exchangeable, we have that,

$$\mathrm{E} d_{\mathrm{TV}}^2\left(\widetilde{P}_{X|Z=Z_i}, P_{X|Z=Z_i}\right) = o(n^{-1}). \tag{D.1.1}$$

This shows that when RHS of (6.2.1), i.e., $\mathrm{E}\{d_{\mathrm{TV}}(\widetilde{P}_{\boldsymbol{X}|\boldsymbol{Z}}, P_{\boldsymbol{X}|\boldsymbol{Z}})\}$ is $o(1)$, (D.1.1) holds.

Next, we show (D.1.1) is violated in the linear regression example. By the data exchangeability, it suffices to show $\sum_{i=1}^n \mathrm{E} d_{\mathrm{TV}}^2\{\widetilde{P}_{X|Z=Z_i}, P_{X|Z=Z_i}\}$ is not $o(1)$. With some calculations, we obtain that,

$$\begin{aligned}
&\sum_{i=1}^n \mathrm{E}\min\left\{1, \sigma_0^{-2}|Z_i^\top(\widehat{\beta}-\beta_0)|^2\right\} \\
&= \sum_{i=1}^n \mathrm{E}\sigma_0^{-2}|Z_i^\top(\widehat{\beta}-\beta_0)|^2 \mathbb{I}\left\{\sigma_0^{-2}|Z_i^\top(\widehat{\beta}-\beta_0)|^2 \le 1\right\} + \sum_{i=1}^n \mathrm{E}\mathbb{I}\left\{\sigma_0^{-2}|Z_i^\top(\widehat{\beta}-\beta_0)|^2 > 1\right\} \\
&= \sum_{i=1}^n \mathrm{E}\sigma_0^{-2}|Z_i^\top(\widehat{\beta}-\beta_0)|^2 - \sum_{i=1}^n \mathrm{E}\left\{\sigma_0^{-2}|Z_i^\top(\widehat{\beta}-\beta_0)|^2 - 1\right\}\mathbb{I}\left\{\sigma_0^{-2}|Z_i^\top(\widehat{\beta}-\beta_0)|^2 > 1\right\}.
\end{aligned} \tag{D.1.2}$$

By the definition of $\widehat{\beta}$, we have

$$\sum_{i=1}^n \mathrm{E}\sigma_0^{-2}|Z_i^\top(\widehat{\beta}-\beta_0)|^2 = \frac{1}{\sigma_0^2}\mathrm{E}(\widehat{\beta}-\beta)^\top \boldsymbol{Z}^\top \boldsymbol{Z}(\widehat{\beta}-\beta) = \frac{1}{\sigma_0^2}\mathrm{E}\boldsymbol{\varepsilon}^\top \boldsymbol{Z}(\boldsymbol{Z}^\top\boldsymbol{Z})^{-1}\boldsymbol{Z}^\top\boldsymbol{\varepsilon},$$

where $\boldsymbol{\varepsilon} = (\varepsilon_1, \cdots, \varepsilon_n)^\top$ consist of i.i.d. copies of $\varepsilon$ defined in Example 1. It follows that,

$$\begin{aligned}
\sum_{i=1}^n \mathrm{E}\sigma_0^{-2}|Z_i^\top(\widehat{\beta}-\beta_0)|^2 = \frac{1}{\sigma_0^2}\mathrm{E}\boldsymbol{\varepsilon}^\top \boldsymbol{Z}(\boldsymbol{Z}^\top\boldsymbol{Z})^\top\boldsymbol{Z}^\top\boldsymbol{\varepsilon} &= \frac{1}{\sigma_0^2}\mathrm{trace}\left\{\mathrm{E}\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^\top \boldsymbol{Z}(\boldsymbol{Z}^\top\boldsymbol{Z})^{-1}\boldsymbol{Z}^\top\right\} \\
&= \mathrm{trace}\left\{\mathrm{E}\boldsymbol{Z}(\boldsymbol{Z}^\top\boldsymbol{Z})^{-1}\boldsymbol{Z}^\top\right\} = d_Z,
\end{aligned} \tag{D.1.3}$$

where $d_Z$ is the dimension of $Z$.

Next, we show that,

$$\sum_{i=1}^{n} \mathrm{E}\sigma_0^{-2}|Z_i^\top(\widehat{\beta}-\beta_0)|^2\mathbb{I}\left\{\sigma_0^{-2}|Z_i^\top(\widehat{\beta}-\beta_0)|^2 \geq 1\right\} = o(1), \tag{D.1.4}$$

or equivalently,

$$\mathrm{E}n\sigma_0^{-2}|Z_i^\top(\widehat{\beta}-\beta_0)|^2\mathbb{I}\left\{\sigma_0^{-2}|Z_i^\top(\widehat{\beta}-\beta_0)|^2 \geq 1\right\} = o(1).$$

We have already shown that $\mathrm{E}n\sigma_0^{-2}|Z_i^\top(\widehat{\beta}-\beta_0)|^2 = d_Z$. By the dominated convergence theorem, it suffices to show that,

$$n\sigma_0^{-2}|Z_i^\top(\widehat{\beta}-\beta_0)|^2\mathbb{I}\left\{\sigma_0^{-2}|Z_i^\top(\widehat{\beta}-\beta_0)|^2 \geq 1\right\} = o_p(1).$$

By definition, it in turn suffices to show that,

$$\Pr\left\{\sigma_0^{-2}|Z_i^\top(\widehat{\beta}-\beta_0)|^2 \geq 1\right\} \to 0.$$

This holds by Markov's inequality, as

$$\mathrm{E}\sigma_0^{-2}|Z_i^\top(\widehat{\beta}-\beta_0)|^2 = \frac{d_Z}{n} \to 0.$$

Combining (D.1.4) together with (D.1.2) and (D.1.3) yields that,

$$\sum_{i=1}^{n} \mathrm{E}\min\left\{1, \sigma_0^{-2}|Z_i^\top(\widehat{\beta}-\beta_0)|^2\right\} \geq d_Z - o(1) \geq 1 - o(1),$$

and hence $\sum_{i=1}^{n} \mathrm{E}d_{\mathrm{TV}}^2\{\widetilde{P}_{X|Z=Z_i}, Q_X^{(n)}(\cdot|Z_i)\} \geq 1 - o(1)$.

This completes the proof of Proposition 2. □

## D.1.2 Proof of Theorem 6.4.2

We begin by providing an upper bound for the function classes $\mathbb{H}_1$ and $\mathbb{H}_2$. Recall that both $\mathbb{H}_1$ and $\mathbb{H}_2$ are classes of neural networks with a single-hidden layer, finitely many hidden nodes, and the sigmoid activation function. Because of that, each function $h_{1,\theta_1} \in \mathbb{H}_1$ and $h_{2,\theta_2} \in \mathbb{H}_2$ can be represented as

$$h_{1,\theta_1}(x) = \sum_{j=1}^{M} \theta_{1,j}^{(1)}\mathrm{sigmoid}(x^\top\theta_{1,j}^{(2)}), \;\; h_{2,\theta_2}(x) = \sum_{j=1}^{M} \theta_{2,j}^{(1)}\mathrm{sigmoid}(y^\top\theta_{2,j}^{(2)}),$$

where $\theta_1$ and $\theta_2$ correspond to the sets of parameters $\{(\theta_{1,j}^{(1)}, \theta_{1,j}^{(2)}) : 1 \le j \le M\}$ and $\{(\theta_{2,j}^{(1)}, \theta_{2,j}^{(2)}) : 1 \le j \le M\}$, respectively, and $M$ is a finite integer. Note that the sigmoid function is bounded. As such, the functions $h_{1,\theta_1}$ and $h_{2,\theta_2}$ are uniformly bounded by $\sum_{j=1}^{M} |\theta_{1,j}^{(1)}|$ and $\sum_{j=1}^{M} |\theta_{2,j}^{(2)}|$, respectively. Since we sample $B$ many functions $\{h_{1,\theta_b}\}_{b=1}^{B}$ and $\{h_{2,\theta_b}\}_{b=1}^{B}$, these functions are uniformly bounded by

$$M \max_{b,j} \left( |\theta_{b,j}^{(1)}| + |\theta_{b,j}^{(2)}| \right).$$

Since these parameters $\theta_1, \theta_2$ are sampled from standard normal distributions, and that

$$\Pr(W > t) = \frac{1}{\sqrt{2\pi}} \int_t^{\infty} \exp\left(-\frac{w^2}{2}\right) dw \le \frac{1}{\sqrt{2\pi}} \int_t^{\infty} w \exp\left(-\frac{w^2}{2}\right) dw = \frac{\exp(-t^2/2)}{\sqrt{2\pi}},$$

for any $t \ge 1$, we can show that $\max_{b,j} \left( |\theta_{b,j}^{(1)}| + |\theta_{b,j}^{(2)}| \right)$ is upper bounded by $\sqrt{\log B}$, with probability approaching one. Note that $B$ grows polynomially with respect to the sample size $n$. Therefore, we have that the functions in $\mathbb{H}_1$ and $\mathbb{H}_2$ are upper bounded by $\log n$ in absolute values.

Define a test statistic

$$T^{**} = \max_{b_1,b_2} \widehat{\sigma}_{b_1,b_2}^{-1} \left| \frac{1}{n} \sum_{i=1}^{n} \left\{ h_{1,b_1}(X_i) - \frac{1}{M} \sum_{m=1}^{M} h_{1,b_1}(X_i^{(m)}) \right\} \left\{ h_{2,b_2}(Y_i) - \frac{1}{M} \sum_{m=1}^{M} h_{2,b_2}(Y_i^{(m)}) \right\} \right|,$$

where the $\widehat{\sigma}_{b_1,b_2}$ is constructed based on $\{\widetilde{X}_i^{(m)}\}_m$ and $\{\widetilde{Y}_i^{(m)}\}_m$, instead of $\{X_i^{(m)}\}_m$ and $\{Y_i^{(m)}\}_m$. It suffices to show that $|\widehat{T} - T^{**}| = O_p(n^{-2\kappa} \log n)$, and $|T^* - T^{**}| = O_p(n^{-2\kappa} \log n)$.

**Step 1.** We first consider the difference $|\widehat{T} - T^{**}|$. For any sequences $\{a_n\}_n$, $\{b_n\}_n$, we have that,

$$|\max_n |a_n| - \max_n |b_n|| \le \max_n |a_n - b_n|. \tag{D.1.5}$$

Consequently, we have $|\widehat{T} - T^{**}| \leq I_1 + I_2 + I_3$, where

$$
\begin{aligned}
I_1 &= \max_{b_1,b_2} \widehat{\sigma}_{b_1,b_2}^{-1} \left| \frac{1}{n}\sum_{i=1}^{n} \left[ \frac{1}{M}\sum_{m=1}^{M} \left\{ h_{1,b_1}(X_i^{(m)}) - h_{1,b_1}(\widetilde{X}_i^{(m)}) \right\} \right] \left\{ h_{2,b_2}(Y_i) - \frac{1}{M}\sum_{m=1}^{M} h_{2,b_2}(Y_i^{(m)}) \right\} \right|, \\
I_2 &= \max_{b_1,b_2} \widehat{\sigma}_{b_1,b_2}^{-1} \left| \frac{1}{n}\sum_{i=1}^{n} \left\{ h_{1,b_1}(X_i) - \frac{1}{M}\sum_{m=1}^{M} h_{1,b_1}(X_i^{(m)}) \right\} \left[ \frac{1}{M}\sum_{m=1}^{M} \left\{ h_{2,b_2}(Y_i^{(m)}) - h_{2,b_2}(\widetilde{Y}_i^{(m)}) \right\} \right] \right|, \\
I_3 &= \max_{b_1,b_2} \widehat{\sigma}_{b_1,b_2}^{-1} \left| \frac{1}{n}\sum_{i=1}^{n} \left[ \frac{1}{M}\sum_{m=1}^{M} \left\{ h_{1,b_1}(X_i^{(m)}) - h_{1,b_1}(\widetilde{X}_i^{(m)}) \right\} \right] \left[ \frac{1}{M}\sum_{m=1}^{M} \left\{ h_{2,b_2}(Y_i^{(m)}) - h_{2,b_2}(\widetilde{Y}_i^{(m)}) \right\} \right] \right|.
\end{aligned}
$$

If $\min \widehat{\sigma}_{b_1,b_2} \geq c_0$ for some constant $c_0 > 0$, then it suffices to show that $I_j^* = O_p(n^{-(\kappa_x+\kappa_y)} \log n)$, for $j = 1,2,3$, where

$$
\begin{aligned}
I_1^* &= \max_{b_1,b_2} \left| \frac{1}{n}\sum_{i=1}^{n} \left[ \frac{1}{M}\sum_{m=1}^{M} \left\{ h_{1,b_1}(X_i^{(m)}) - h_{1,b_1}(\widetilde{X}_i^{(m)}) \right\} \right] \left\{ h_{2,b_2}(Y_i) - \frac{1}{M}\sum_{m=1}^{M} h_{2,b_2}(Y_i^{(m)}) \right\} \right|, \\
I_2^* &= \max_{b_1,b_2} \left| \frac{1}{n}\sum_{i=1}^{n} \left\{ h_{1,b_1}(X_i) - \frac{1}{M}\sum_{m=1}^{M} h_{1,b_1}(X_i^{(m)}) \right\} \left[ \frac{1}{M}\sum_{m=1}^{M} \left\{ h_{2,b_2}(Y_i^{(m)}) - h_{2,b_2}(\widetilde{Y}_i^{(m)}) \right\} \right] \right|, \\
I_3^* &= \max_{b_1,b_2} \left| \frac{1}{n}\sum_{i=1}^{n} \left[ \frac{1}{M}\sum_{m=1}^{M} \left\{ h_{1,b_1}(X_i^{(m)}) - h_{1,b_1}(\widetilde{X}_i^{(m)}) \right\} \right] \left[ \frac{1}{M}\sum_{m=1}^{M} \left\{ h_{2,b_2}(Y_i^{(m)}) - h_{2,b_2}(\widetilde{Y}_i^{(m)}) \right\} \right] \right|.
\end{aligned}
$$

The number of folds $L$ is finite, as such, it suffices to show that $I_j^{(\ell)} = O_p(n^{-(\kappa_x+\kappa_y)} \log n)$, for $j = 1,2,3$ and $\ell = 1,\ldots,L$, where

$$
\begin{aligned}
I_1^{(\ell)} &= \max_{b_1,b_2} \left| \frac{1}{n}\sum_{i\in\mathcal{I}^{(\ell)}} \left[ \frac{1}{M}\sum_{m=1}^{M} \left\{ h_{1,b_1}(X_i^{(m)}) - h_{1,b_1}(\widetilde{X}_i^{(m)}) \right\} \right] \left\{ h_{2,b_2}(Y_i) - \frac{1}{M}\sum_{m=1}^{M} h_{2,b_2}(Y_i^{(m)}) \right\} \right|, \\
I_2^{(\ell)} &= \max_{b_1,b_2} \left| \frac{1}{n}\sum_{i\in\mathcal{I}^{(\ell)}} \left\{ h_{1,b_1}(X_i) - \frac{1}{M}\sum_{m=1}^{M} h_{1,b_1}(X_i^{(m)}) \right\} \left[ \frac{1}{M}\sum_{m=1}^{M} \left\{ h_{2,b_2}(Y_i^{(m)}) - h_{2,b_2}(\widetilde{Y}_i^{(m)}) \right\} \right] \right|, \\
I_3^{(\ell)} &= \max_{b_1,b_2} \left| \frac{1}{n}\sum_{i\in\mathcal{I}^{(\ell)}} \left[ \frac{1}{M}\sum_{m=1}^{M} \left\{ h_{1,b_1}(X_i^{(m)}) - h_{1,b_1}(\widetilde{X}_i^{(m)}) \right\} \right] \left[ \frac{1}{M}\sum_{m=1}^{M} \left\{ h_{2,b_2}(Y_i^{(m)}) - h_{2,b_2}(\widetilde{Y}_i^{(m)}) \right\} \right] \right|.
\end{aligned}
$$

We divide the rest of the proof into four sub-steps. We first show that $I_j^{(\ell)} = O_p(n^{-(\kappa_x+\kappa_y)} \log n)$, for $j = 1,2,3$. Finally, we show $\Pr(\min \widehat{\sigma}_{b_1,b_2} \geq c_0) \to 1$ for some constant $c_0 > 0$.

**Step 1.1.** Recall we have shown that the functions in $\mathbb{H}_1$ and $\mathbb{H}_2$ are bounded by $\log n$ in absolute values at the beginning of the proof of Theorem 6.4.2. By Bernstein's inequality,

we have that,

$$\Pr\left[\left|\sum_{m=1}^{M} h_{1,b}(X_i^{(m)}) - M\mathrm{E}\{h_{1,b}(X_i)|Z_i\}\right| \geq t\right] \leq 2\exp\left\{-\frac{t^2}{2(M\log n + t\sqrt{\log n}/3)}\right\},$$

for any $b$ and $i$. Set $t = \sqrt{3(c+2)M}\log n$, where the constant $c$ is as defined in the statement of Theorem 6.4.1. For a sufficiently large $n$, we have $t\sqrt{\log n}/3 \leq M\log n/2$. It follows that

$$\Pr\left[\left|\sum_{m=1}^{M} h_{1,b}(X_i^{(m)}) - M\mathrm{E}\{h_{1,b}(X_i)|Z_i\}\right| \geq \sqrt{3(c+2)M}\log n\right] \leq \frac{2}{n^{c+2}}.$$

By Bonferroni's inequality, we obtain that,

$$\begin{aligned}
&\Pr\left[\max_{b\in\{1,\cdots,B\}}\max_{i\in\{1,\cdots,n\}}\left|\sum_{m=1}^{M} h_{1,b}(X_i^{(m)}) - M\mathrm{E}\{h_{1,b}(X_i)|Z_i\}\right| \geq \sqrt{3(c+2)M}\log n\right] \\
\leq\ & Bn\max_{b\in\{1,\cdots,B\}}\max_{i\in\{1,\cdots,n\}}\Pr\left[\left|\sum_{m=1}^{M} h_{1,b}(X_i^{(m)}) - M\mathrm{E}\{h_{1,b}(X_i)|Z_i\}\right| \geq \sqrt{3(c+2)M}\log n\right] \leq \frac{2Bn}{n^{c+2}}.
\end{aligned}$$

Under the condition $B = O(n^c)$, we obtain with probability $1 - O(n^{-1})$ that,

$$\max_{b\in\{1,\cdots,B\}}\max_{i\in\{1,\cdots,n\}}\left|\sum_{m=1}^{M} h_{1,b}(X_i^{(m)}) - M\mathrm{E}\{h_{1,b}(X_i)|Z_i\}\right| \leq O(1)n^{-1/2}\log n, \qquad \text{(D.1.6)}$$

as $M$ is proportional to $n$, and $O(1)$ denotes some positive constant.

Similarly, we can show that,

$$\max_{b\in\{1,\cdots,B\}}\max_{i\in\mathcal{I}^{(\ell)}}\left|\sum_{m=1}^{M} h_{1,b}(\widetilde{X}_i^{(m)}) - M\int_x h_{1,b}(x)\widetilde{P}^{(\ell)}_{X|Z=Z_i}(dx)\right| \leq O(1)\sqrt{n}\log n,$$

with probability $1 - O(n^{-1})$. Combining this with (D.1.6), we obtain with probability $1 - O(n^{-1})$ that,

$$\begin{aligned}
\max_{\substack{b\in\{1,\ldots,B\}\\ i\in\mathcal{I}^{(\ell)}}}\Bigg|&\sum_{m=1}^{M}\left\{h_{1,b}(X_i^{(m)}) - h_{1,b}(\widetilde{X}_i^{(m)})\right\} \\
&-M\int_x h_{1,b}(x)\left\{P_{X|Z=Z_i}(dx) - \widetilde{P}^{(\ell)}_{X|Z=Z_i}(dx)\right\}\Bigg| \leq O(1)\sqrt{n}\log n.
\end{aligned} \qquad \text{(D.1.7)}$$

Conditional on $Z_i$, the expectation of $h_{2,b_2}(Y_i) - M^{-1}\sum_{m=1}^{M} h_{2,b_2}(Y_i^{(m)})$ equals zero. Under the null hypothesis, the expectation of $M^{-1}\sum_{m=1}^{M}\{h_{1,b_1}(X_i^{(m)}) - h_{1,b_1}(\widetilde{X}_i^{(m)})\}\{h_{2,b_2}(Y_i) - M^{-1}\sum_{m=1}^{M} h_{2,b_2}(Y_i^{(m)})\}$ equals zero as well. Applying Bernstein's inequality again, we can show with probability tending to 1 that,

$$I_1^{(\ell)} \leq O(1)\left(\sigma n^{-1/2}\log^{3/2} n + n^{-1}\log^2 n\right), \tag{D.1.8}$$

where

$$\begin{aligned}\sigma^2 = \max_{b_1,b_2} \mathrm{E}\left|\frac{1}{M}\sum_{m=1}^{M}\left\{h_{1,b_1}(X_i^{(m)}) - h_{1,b_1}(\widetilde{X}_i^{(m)})\right\}\left\{h_{2,b_2}(Y_i) - \frac{1}{M}\sum_{m=1}^{M} h_{2,b_2}(Y_i^{(m)})\right\}\right|^2 \\ \leq \max_{b_1} \mathrm{E}\left|\frac{1}{M}\sum_{m=1}^{M}\left\{h_{1,b_1}(X_i^{(m)}) - h_{1,b_1}(\widetilde{X}_i^{(m)})\right\}\right|^2 \log n.\end{aligned}$$

Let $\mathcal{A}$ denote the event in (D.1.7). The last term on the second line can be bounded from above by

$$\max_{b_1,i} \mathrm{E}\left|\frac{1}{M}\sum_{m=1}^{M}\{h_{1,b_1}(X_i^{(m)}) - h_{1,b_1}(\widetilde{X}_i^{(m)})\}\right|^2 \mathbb{I}(\mathcal{A})\log n \tag{D.1.9}$$

$$+ \quad \max_{b_1,i} \mathrm{E}\left|\frac{1}{M}\sum_{m=1}^{M}\{h_{1,b_1}(X_i^{(m)}) - h_{1,b_1}(\widetilde{X}_i^{(m)})\}\right|^2 \mathbb{I}(\mathcal{A}^c)\log n. \tag{D.1.10}$$

Since $M$ is proportional to $n$, by (D.1.5), (D.1.9) is upper bounded by

$$O(1)\left[n^{-1}\log^2 n + \max_{\substack{b\in\{1,\cdots,B\}\\ i\in\mathcal{I}^{(\ell)}}} \mathrm{E}\left|\int_x h_{1,b}(x)\left\{\widetilde{P}_{X|Z=Z_i}^{(\ell)}(dx) - P_{X|Z=Z_i}(dx)\right\}\right|^2\right]\log n.$$

By the boundedness of the function class $\mathbb{H}_1$, it can be further bounded from above by

$$O(1)\left\{n^{-1}\log^3 n + \mathrm{E}d_{\mathrm{TV}}^2(\widetilde{P}_{X|Z}^{(\ell)}, P_{X|Z})\log^2 n\right\}. \tag{D.1.11}$$

The above quantity is of order $O(n^{-2\kappa_x}\log^2 n)$. Consequently, (D.1.9) is of the order $O(n^{-2\kappa_x}\log^2 n)$.

Note that the event $\mathcal{A}$ occurs with probability at least $1 - O(n^{-1})$. By the boundedness of the function class $\mathbb{H}_1$, (D.1.10) is of the order $O(n^{-1}\log^2 n)$.

Therefore, $\sigma^2$ is of the order $O(n^{-2\kappa_x}\log^2 n)$. This implies that $\mathcal{I}_1^{(\ell)}$ can be bounded from above by $O(n^{-1/2-\kappa_x}\log^{5/2} n)$, which in turn yields that $\mathcal{I}_1^{(\ell)} = O_p(n^{-\kappa_x-\kappa_y}\log n)$, since $\kappa_x, \kappa_y < 1/2$.

**Step 1.2.** This step can be proven in a similar way as Step 1.1, and is omitted.

**Step 1.3.** Under $H_0$, the expectation of

$$\frac{1}{|\mathcal{I}^{(\ell)}|}\sum_{i\in\mathcal{I}^{(\ell)}}\left[\frac{1}{M}\sum_{m=1}^{M}\left\{h_{1,b_1}(X_i^{(m)}) - h_{1,b_1}(\widetilde{X}_i^{(m)})\right\}\right]\left[\frac{1}{M}\sum_{m=1}^{M}\left\{h_{2,b_2}(Y_i^{(m)}) - h_{2,b_2}(\widetilde{Y}_i^{(m)})\right\}\right]$$

equals

$$\mathrm{E}\int_x h_{1,b_1}(x)\left\{\widetilde{P}^{(\ell)}_{X|Z}(dx) - P_{X|Z}(dx)\right\}\int_y h_{2,b_2}(y)\left\{\widetilde{P}^{(\ell)}_{Y|Z}(dy) - P_{Y|Z}(dy)\right\}.$$

Similar to (D.1.11), its absolute value can be upper bounded by

$$\mathrm{E}d_{\mathrm{TV}}\left\{\widetilde{P}^{(\ell)}_{X|Z=Z_i}, P_{X|Z}\right\} d_{\mathrm{TV}}\left\{\widetilde{P}^{(\ell)}_{Y|Z=Z_i}, P_{Y|Z}\right\}\log n.$$

Following Cauchy-Schwarz inequality, we have that,

$$\begin{aligned}&\mathrm{E}d_{\mathrm{TV}}\left\{\widetilde{P}^{(\ell)}_{X|Z=Z_i}, P_{X|Z}\right\} d_{\mathrm{TV}}\left\{\widetilde{P}^{(\ell)}_{Y|Z=Z_i}, P_{Y|Z}\right\}\\ \leq\ &\sqrt{\mathrm{E}d^2_{\mathrm{TV}}\left\{\widetilde{P}^{(\ell)}_{X|Z=Z_i}, P_{X|Z}\right\}\mathrm{E}d^2_{\mathrm{TV}}\left\{\widetilde{P}^{(\ell)}_{Y|Z=Z_i}, P_{Y|Z}\right\}} = O(n^{-(\kappa_x+\kappa_y)}).\end{aligned}$$

This yields that,

$$\max_{b_1,b_2}\left|\mathrm{E}\int_x h_{1,b_1}(x)\left\{\widetilde{P}^{(\ell)}_{X|Z}(dx) - P_{X|Z}(dx)\right\}\int_y h_{2,b_2}(y)\left\{\widetilde{P}^{(\ell)}_{Y|Z}(dy) - P_{Y|Z}(dy)\right\}\right| = O(n^{-(\kappa_x+\kappa_y)}\log n).$$

Following similar arguments as in Step 1.1, we obtain that,

$$\begin{aligned}I_3^{(\ell)} - \max_{b_1,b_2}\left|\mathrm{E}\int_x h_{1,b_1}(x)\left\{\widetilde{P}^{(\ell)}_{X|Z}(dx) - P_{X|Z}(dx)\right\}\int_y h_{2,b_2}(y)\left\{\widetilde{P}^{(\ell)}_{Y|Z}(dy) - P_{Y|Z}(dy)\right\}\right|&\\ = O_p(n^{-(\kappa_x+\kappa_y)}\log n).&\end{aligned}$$

Therefore, we obtain that $I_3^{(\ell)} = O_p(n^{-(\kappa_x+\kappa_y)}\log n)$.

**Step 1.4.** Recall that $\widehat{\sigma}^2_{b_1,b_2}$ is defined by

$$\frac{1}{n-1}\sum_{i=1}^{n}\left(\left[h_{1,b_1}(X_i) - \widehat{\mathrm{E}}\{h_{1,b_1}(X_i)|Z_i\}\right]\left[h_{2,b_2}(Y_i) - \widehat{\mathrm{E}}\{h_{2,b_2}(Y_i)|Z_i\}\right] - \mathrm{GCM}\{h_{1,b_1}(X), h_{2,b_2}(Y)\}\right)^2.$$

With some calculations, it is equal to

$$\frac{1}{n-1}\sum_{i=1}^{n}\left[h_{1,b_1}(X_i)-\widehat{\mathrm{E}}\{h_{1,b_1}(X_i)|Z_i\}\right]^2\left[h_{2,b_2}(Y_i)-\widehat{\mathrm{E}}\{h_{2,b_2}(Y_i)|Z_i\}\right]^2 - \frac{n}{n-1}\mathrm{GCM}^2\{h_{1,b_1}(X),h_{2,b_2}(Y)\}, \tag{D.1.12}$$

where the estimated conditional expectation $\widehat{\mathbb{E}}$ is computed using GANs.

Consider the second term $\mathrm{GCM}\{h_{1,b_1}(X),h_{2,b_2}(Y)\}$ in (D.1.12). Following similar arguments as in Steps 1.1 and 1.3, we have that,

$$\max_{b_1,b_2}\left|\mathrm{GCM}\{h_{1,b_1}(X),h_{2,b_2}(Y)\}-\mathrm{GCM}'\{h_{1,b_1}(X),h_{2,b_2}(Y)\}\right| = O_p(n^{-(\kappa_x+\kappa_y)}\log n),$$

where $\mathrm{GCM}'\{h_{1,b_1}(X),h_{2,b_2}(Y)\}$ equals

$$\frac{1}{n}\sum_{i=1}^{n}\left\{h_{1,b_1}(X_i)-\frac{1}{M}\sum_{m=1}^{M}h_{1,b_1}(X_i^{(m)})\right\}\left\{h_{2,b_2}(Y_i)-\frac{1}{M}\sum_{m=1}^{M}h_{2,b_2}(Y_i^{(m)})\right\}.$$

Similar to (D.1.7), we can show that,

$$\max_{b_1,b_2}\left|\mathrm{GCM}'\{h_{1,b_1}(X),h_{2,b_2}(Y)\}-\mathrm{GCM}^*\{h_{1,b_1}(X),h_{2,b_2}(Y)\}\right| = O_p\left(n^{-1/2}\sqrt{\log n}\right).$$

Consequently, we have that,

$$\max_{b_1,b_2}\left|\mathrm{GCM}\{h_{1,b_1}(X),h_{2,b_2}(Y)\}-\mathrm{GCM}^*\{h_{1,b_1}(X),h_{2,b_2}(Y)\}\right| = O_p\left(n^{-1/2}\sqrt{\log n}\right).$$

Since the function classes $\mathbb{H}_1$ and $\mathbb{H}_2$ are bounded, both GCM and $\mathrm{GCM}^*$ are bounded by $\log n$ in absolute values. Consequently,

$$\max_{b_1,b_2}\left|\mathrm{GCM}^2\{h_{1,b_1}(X),h_{2,b_2}(Y)\}-\mathrm{GCM}^{*2}\{h_{1,b_1}(X),h_{2,b_2}(Y)\}\right| = O_p\left(n^{-1/2}\log^{3/2}n\right). \tag{D.1.13}$$

Next, consider the first term in (D.1.12). Note that it can be represented by

$$\frac{n}{n-1}\frac{1}{L}\sum_{\ell=1}^{L}\left(\frac{1}{|\mathcal{I}^\ell|}\sum_{i\in\mathcal{I}^{(\ell)}}\left[h_{1,b_1}(X_i)-\widehat{\mathrm{E}}\{h_{1,b_1}(X_i)|Z_i\}\right]^2\left[h_{2,b_2}(Y_i)-\widehat{\mathrm{E}}\{h_{2,b_2}(Y_i)|Z_i\}\right]^2\right).$$

Similar to (D.1.7), we can show that,

$$\max_{b_1,b_2}\Bigg|\frac{1}{|\mathcal{I}^{\ell}|}\sum_{i\in\mathcal{I}^{(\ell)}}\Big[h_{1,b_1}(X_i)-\widehat{\mathrm{E}}\{h_{1,b_1}(X_i)|Z_i\}\Big]^2\Big[h_{2,b_2}(Y_i)-\widehat{\mathrm{E}}\{h_{2,b_2}(Y_i)|Z_i\}\Big]^2$$
$$-\,\mathrm{E}\Big[h_{1,b_1}(X_1)-\widehat{\mathrm{E}}\{h_{1,b_1}(X_1)|Z_1\}\Big]^2\Big[h_{2,b_2}(Y_1)-\widehat{\mathrm{E}}\{h_{2,b_2}(Y_1)|Z_1\}\Big]^2\Bigg| = O_p(n^{-1/2}\log^{3/2}n).$$

Following similar arguments as in Steps 1.1 and 1.3, we can show that,

$$\max_{b_1,b_2}\Bigg|\mathrm{E}\Big[h_{1,b_1}(X_1)-\widehat{\mathrm{E}}\{h_{1,b_1}(X_1)|Z_1\}\Big]^2\Big[h_{2,b_2}(Y_1)-\widehat{\mathrm{E}}\{h_{2,b_2}(Y_1)|Z_1\}\Big]^2$$
$$-\,\mathrm{E}\big[h_{1,b_1}(X_1)-\mathrm{E}\{h_{1,b_1}(X_1)|Z_1\}\big]^2\big[h_{2,b_2}(Y_1)-\mathrm{E}\{h_{2,b_2}(Y_1)|Z_1\}\big]^2\Bigg| = O_p(n^{-\bar{c}}),$$

for some constant $0<\bar{c}<1/2$. It follows that,

$$\max_{b_1,b_2}\Bigg|\frac{1}{|\mathcal{I}^{\ell}|}\sum_{i\in\mathcal{I}^{(\ell)}}\Big[h_{1,b_1}(X_i)-\widehat{\mathrm{E}}\{h_{1,b_1}(X_i)|Z_i\}\Big]^2\Big[h_{2,b_2}(Y_i)-\widehat{\mathrm{E}}\{h_{2,b_2}(Y_i)|Z_i\}\Big]^2$$
$$-\,\mathrm{E}\big[h_{1,b_1}(X)-\mathrm{E}\{h_{1,b_1}(X)|Z\}\big]^2\big[h_{2,b_2}(Y)-\mathrm{E}\{h_{2,b_2}(Y)|Z\}\big]^2\Bigg| = O_p(n^{-\bar{c}}),$$

and henceforth,

$$\max_{b_1,b_2}\Bigg|\frac{1}{n}\sum_{i=1}^{n}\Big[h_{1,b_1}(X_i)-\widehat{\mathrm{E}}\{h_{1,b_1}(X_i)|Z_i\}\Big]^2\Big[h_{2,b_2}(Y_i)-\widehat{\mathrm{E}}\{h_{2,b_2}(Y_i)|Z_i\}\Big]^2$$
$$-\,\mathrm{E}\big[h_{1,b_1}(X)-\mathrm{E}\{h_{1,b_1}(X)|Z\}\big]^2\big[h_{2,b_2}(Y)-\mathrm{E}\{h_{2,b_2}(Y)|Z\}\big]^2\Bigg| = O_p(n^{-\bar{c}}).$$

Combining this together with (D.1.13) yields that,

$$\max_{b_1,b_2}\Bigg|\widehat{\sigma}^2_{b_1,b_2}-\frac{n}{n-1}\mathrm{Var}\Big(\big[h_{1,b_1}(X)-\mathrm{E}\{h_{1,b_1}(X)|Z\}\big]\big[h_{2,b_2}(Y)-\mathrm{E}\{h_{2,b_2}(Y)|Z\}\big]\Big)\Bigg| = O_p(n^{-\bar{c}}).$$

Then, we have that,

$$\min_{b_1,b_2}\mathrm{Var}\Big(\big[h_{1,b_1}(X)-\mathrm{E}\{h_{1,b_1}(X)|Z\}\big]\big[h_{2,b_2}(Y)-\mathrm{E}\{h_{2,b_2}(Y)|Z\}\big]\Big)\geq c^*,$$

for some constant $c^*>0$. Therefore, we have that

$$\min_{b_1,b_2}\widehat{\sigma}^2_{b_1,b_2}\geq 2^{-1}c^*,$$

with probability tending to 1.

**Step 2.** We next consider the difference $|T^* - T^{**}|$, and show that it is of the order $O_p(n^{-2\kappa}\log n)$. Denote by $\widehat{\sigma}^{*2}_{b_1,b_2}$ the variance estimator with $\{\widetilde{X}_i^{(m)}\}_m$ and $\{\widetilde{Y}_i^{(m)}\}_m$ replaced by $\{X_i^{(m)}\}_m$ and $\{Y_i^{(m)}\}_m$. Using (D.1.5), the difference between $T^*$ and $T^{**}$ is upper bounded by

$$\max_{b_1,b_2} |\widehat{\sigma}^{-1}_{b_1,b_2} - \widehat{\sigma}^{*-1}_{b_1,b_2}| \left| \frac{1}{n}\sum_{i=1}^{n} \left\{ h_{1,b_1}(X_i) - \frac{1}{M}\sum_{m=1}^{M} h_{1,b_1}(X_i^{(m)}) \right\} \left\{ h_{2,b_2}(Y_i) - \frac{1}{M}\sum_{m=1}^{M} h_{2,b_2}(Y_i^{(m)}) \right\} \right|.$$

Under $H_0$, similar to (D.1.7), we can show that,

$$\max_{b_1,b_2} \left| \frac{1}{n}\sum_{i=1}^{n} \left\{ h_{1,b_1}(X_i) - \frac{1}{M}\sum_{m=1}^{M} h_{1,b_1}(X_i^{(m)}) \right\} \left\{ h_{2,b_2}(Y_i) - \frac{1}{M}\sum_{m=1}^{M} h_{2,b_2}(Y_i^{(m)}) \right\} \right| = O_p(n^{-1/2}\log^{3/2} n).$$

To show $|T^* - T^{**}| = O_p(n^{-2\kappa}\log n)$, it suffices to show that $\max_{b_1,b_2} |\widehat{\sigma}^{-1}_{b_1,b_2} - \widehat{\sigma}^{*-1}_{b_1,b_2}| = O_p(n^{-\bar{c}})$ for some constant $\bar{c} > 0$. Since both $\widehat{\sigma}^{-1}_{b_1,b_2}$ and $\widehat{\sigma}_{b_1,b_2}$ are bounded away from zero, it suffices to show that $\max_{b_1,b_2} |\widehat{\sigma}^{2}_{b_1,b_2} - \widehat{\sigma}^{*2}_{b_1,b_2}| = O_p(n^{-\bar{c}})$.

Following similar arguments as in Steps 1.1 and 1.3, we can show that,

$$\max_{b_1,b_2} \left| \widehat{\sigma}^{2}_{b_1,b_2} - \frac{n}{n-1}\mathrm{Var}\Big( \big[h_{1,b_1}(X) - \mathrm{E}\{h_{1,b_1}(X)|Z\}\big] \big[h_{2,b_2}(Y) - \mathrm{E}\{h_{2,b_2}(Y)|Z\}\big] \Big) \right| = O_p(n^{-\bar{c}}),$$
$$\max_{b_1,b_2} \left| \widehat{\sigma}^{*2}_{b_1,b_2} - \frac{n}{n-1}\mathrm{Var}\Big( \big[h_{1,b_1}(X) - \mathrm{E}\{h_{1,b_1}(X)|Z\}\big] \big[h_{2,b_2}(Y) - \mathrm{E}\{h_{2,b_2}(Y)|Z\}\big] \Big) \right| = O_p(n^{-\bar{c}}).$$

This completes the proof of Theorem 6.4.2. □

## D.1.3 Proof of Theorem 6.4.3

In the proof of Theorem 6.4.2, we have already shown that $\widehat{T} - T^* = O_p(n^{-(\kappa_x+\kappa_y)}\log n)$. Following similar arguments as in Step 1.4, we can show that $T^* - T^{***} = O_p(n^{-(\kappa_x+\kappa_y)}\log n)$, where

$$T^{***} = \max_{b_1,b_2} \sigma^{-1}_{b_1,b_2} | \left| n^{-1}\sum_{i=1}^{n} \left\{ h_{1,b_1}(X_i) - \frac{1}{M}\sum_{m=1}^{M} h_{1,b_1}(X_i^{(m)}) \right\} \left\{ h_{2,b_2}(Y_i) - \frac{1}{M}\sum_{m=1}^{M} h_{2,b_2}(Y_i^{(m)}) \right\} \right|,$$

where

$$\sigma^{2}_{b_1,b_2} = \frac{n}{n-1}\mathrm{Var}\Big( \big[h_{1,b_1}(X) - \mathrm{E}\{h_{1,b_1}(X)|Z\}\big] \big[h_{2,b_2}(Y) - \mathrm{E}\{h_{2,b_2}(Y)|Z\}\big] \Big).$$

By (D.1.6), following similar arguments as in the proof regarding the term $I_1$ in Theorem 6.4.2, we can show that $T^{***} - T^{****} = O_p(n^{-(\kappa_x+\kappa_y)}\log n)$, where

$$T^{****} = \max_{b_1,b_2} \sigma_{b_1,b_2}^{-1} \left| n^{-1} \sum_{i=1}^{n} \left[ h_{1,b_1}(X_i) - \mathrm{E}\{h_{1,b_1}(X_i)|Z_i\} \right] \left\{ h_{2,b_2}(Y_i) - \frac{1}{M} \sum_{m=1}^{M} h_{2,b_2}(Y_i^{(m)}) \right\} \right|.$$

Similarly, we can show that $T^{****} - T_0 = O_p(n^{-(\kappa_x+\kappa_y)}\log n)$, where

$$T_0 = \max_{b_1,b_2} \sigma_{b_1,b_2}^{-1} \left| n^{-1} \sum_{i=1}^{n} \left[ h_{1,b_1}(X_i) - \mathrm{E}\{h_{1,b_1}(X_i)|Z_i\} \right] \left[ h_{2,b_2}(Y_i) - \mathrm{E}\{h_{2,b_2}(Y_i)|Z_i\} \right] \right|.$$

Therefore, we have shown that $\widehat{T} - T_0 = O_p(n^{-(\kappa_x+\kappa_y)}\log n)$. Since $\kappa_x + \kappa_y > 1/2$, we have that,

$$\sqrt{n}(\widehat{T} - T_0) = o_p(\log^{-1/2} n). \tag{D.1.14}$$

Define a $B^2 \times B^2$ matrix $\Sigma_0$ whose $\{b_1 + B(b_2-1), b_3 + B(b_4-1)\}$th entry is given by

$$\begin{aligned} \mathrm{cov} \Big( & \sigma_{b_1,b_2}^{-1} \left[ h_{1,b_1}(X_i) - \mathrm{E}\{h_{1,b_1}(X_i)|Z_i\} \right] \left[ h_{2,b_2}(Y_i) - \mathrm{E}\{h_{2,b_2}(Y_i)|Z_i\} \right], \\ & \sigma_{b_3,b_4}^{-1} \left[ h_{1,b_3}(X_i) - \mathrm{E}\{h_{1,b_3}(X_i)|Z_i\} \right] \left[ h_{2,b_4}(Y_i) - \mathrm{E}\{h_{2,b_4}(Y_i)|Z_i\} \right] \Big). \end{aligned}$$

In the following, we show that,

$$\sup_t \left| \Pr\left( \sqrt{n}\widehat{T}_0 \le t | \mathcal{H}_0 \right) - \Pr\left( \|N(0,\Sigma_0)\|_\infty \le t \right) \right| = o(1). \tag{D.1.15}$$

When $B$ is finite, this is implied by the classical weak convergence results. When $B$ diverges with $n$, we require $B = O(n^c)$ for some constant $c > 0$. By the definition of $\sigma_{b_1,b_2}$, the variance of

$$\sigma_{b_1,b_2}^{-1} \left[ h_{1,b_1}(X_i) - \mathrm{E}\{h_{1,b_1}(X_i)|Z_i\} \right] \left[ h_{2,b_2}(Y_i) - \mathrm{E}\{h_{2,b_2}(Y_i)|Z_i\} \right]$$

is bounded from above by $(n-1)/n$. Moreover, combining the boundedness of the function spaces $\mathbb{H}_1$ and $\mathbb{H}_2$ together with the definition of $\sigma_{b_1,b_2}$ yields that,

$$\left\{ \sigma_{b_1,b_2}^{-1} \left[ h_{1,b_1}(X_i) - \mathrm{E}\{h_{1,b_1}(X_i)|Z_i\} \right] \left[ h_{2,b_2}(Y_i) - \mathrm{E}\{h_{2,b_2}(Y_i)|Z_i\} \right] : b_1, b_2 \in \{1, \cdots, B\} \right\}$$

are uniformly bounded from infinity by $O(\log n)$, with probability tending to 1. We can show that (D.1.15) holds. This implies that,

$$\sigma_{b_1,b_2}^{-1} n^{-1/2} \sum_{i=1}^{n} \left[h_{1,b_1}(X_i) - \mathrm{E}\{h_{1,b_1}(X_i)|Z_i\}\right] \left[h_{2,b_2}(Y_i) - \mathrm{E}\{h_{2,b_2}(Y_i)|Z_i\}\right]$$

is asymptotically normal with zero mean.

Combining (D.1.15) together with (D.1.14) yields that,

$$\begin{aligned} \Pr\left(\sqrt{n}\widehat{T} \leq t|\mathcal{H}_0\right) &\geq \Pr\left(\|N(0,\Sigma_0)\|_\infty \leq t - \varepsilon_0 \log^{-1/2} n\right) - o(1), \\ \Pr\left(\sqrt{n}\widehat{T} \leq t|\mathcal{H}_0\right) &\leq \Pr\left(\|N(0,\Sigma_0)\|_\infty \leq t + \varepsilon_0 \log^{-1/2} n\right) + o(1), \end{aligned} \tag{D.1.16}$$

for any sufficiently small $\varepsilon_0 > 0$, where the little-o terms are uniform in $t$.

Following similar arguments as in Step 1.4 and Step 2 of the proof of Theorem 6.4.2, we can show that $\|\widehat{\Sigma} - \Sigma_0\|_{\infty,\infty} = O_p(n^{-\bar{c}})$ for some constant $\bar{c} > 0$. Following similar arguments for (D.1.16), we have that,

$$\begin{aligned} \Pr\left(\sqrt{n}\widehat{T} \leq t|\mathcal{H}_0\right) &\geq \Pr\left(\|N(0,\widehat{\Sigma})\|_\infty \leq t - 2\varepsilon_0 \log^{-1/2} n|\widehat{\Sigma}\right) - o(1), \\ \Pr\left(\sqrt{n}\widehat{T} \leq t|\mathcal{H}_0\right) &\leq \Pr\left(\|N(0,\widehat{\Sigma})\|_\infty \leq t + 2\varepsilon_0 \log^{-1/2} n|\widehat{\Sigma}\right) + o(1), \end{aligned}$$

for any sufficiently small $\varepsilon_0 > 0$. Since the little-o terms are uniform in $t \in \mathbb{R}$, we obtain that,

$$\begin{aligned} &\sup_t |\Pr(\sqrt{n}\widehat{T} \leq t|\mathcal{H}_0) - \Pr(\|N(0,\widehat{\Sigma})\|_\infty \leq t|\widehat{\Sigma})| \leq o(1) \\ + \quad &\sup_t |\Pr(\|N(0,\widehat{\Sigma})\|_\infty \leq t + 2\varepsilon \log^{-1/2} n|\widehat{\Sigma}) - \Pr(\|N(0,\widehat{\Sigma})\|_\infty \leq t - 2\varepsilon_0 \log^{-1/2} n|\widehat{\Sigma})|. \end{aligned}$$

By Theorem 1 of [33], the term on the second line can be bounded by $O(1)\varepsilon_0 \log^{1/2} B \log^{-1/2} n$, where $O(1)$ denotes some positive constant. Since $B = O(n^c)$, $\log^{1/2} B \log^{-1/2} n = O(1)$. As $\varepsilon_0$ grows to zero, this term becomes negligible. Consequently, we obtain that,

$$\sup_t \left|\Pr\left(\sqrt{n}\widehat{T} \leq t|\mathcal{H}_0\right) - \Pr\left(\|N(0,\widehat{\Sigma})\|_\infty \leq t|\widehat{\Sigma}\right)\right| \leq o(1).$$

As such, the distribution of our test statistic can be well-approximated by that of the bootstrap samples. This completes the proof of Theorem 6.4.3. □

### D.1.4 Proof of Theorem 6.4.4

We break the proof into two steps. In Step 1, we show that, under $\mathcal{H}_1^*$, there exist two neural networks functions $f(X) \in \mathbb{H}_1$ and $g(Y) \in \mathbb{H}_2$, such that

$$I(f,g) = \mathrm{E}[f(X) - \mathrm{E}\{f(X)|Z\}][g(Y) - \mathrm{E}\{g(Y)|Z\}] \neq 0,$$

In Step 2, we prove the power of our test approaches one, as the sample size diverges to infinity.

**Step 1.** We first observe that the measure $I(f,g) = \mathrm{E}[f(X) - \mathrm{E}\{f(X)|Z\}][g(Y) - \mathrm{E}\{g(Y)|Z\}]$ is continuous in $f$ and $g$. That is, for any $f_1, f_2 \in L_X^2$ and $g_1, g_2 \in L_Y^2$, the difference $I(f_1,g_1) - I(f_2,g_2)$ decays to zero as both $\mathrm{E}|f_1(X) - f_2(X)|^2$ and $\mathrm{E}|g_1(X) - g_2(X)|^2$ decay to zero.

Under $\mathcal{H}_1^*$, there exist functions $f^* \in L_X^2$ and $g^* \in L_Y^2$, such that $I(f^*,g^*) \neq 0$. Without loss of generality, assume $f^*$ and $g^*$ are bounded. Otherwise, we can find sequences of bounded functions $\{f_n^*\}_n$ and $\{g_n^*\}_n$ that converge to $f^*$ and $g^*$ under $L_2$-norm, respectively. As a result, we would have $I(f_n^*, g_n^*) \neq 0$ for some $n$.

By Lusin's theorem, we can find a sequence of bounded and continuous functions $\{f_n^{**}\}_n$, such that $\lim_n \Pr(f_n^{**}(X) \neq f^*(X)) = 0$. By dominated convergence theorem, it follows that $f_n^{**}$ converges to $f^*$ under $L_2$-norm. Similarly, we can find a sequence of continuous functions $\{g_n^{**}\}_n$, such that $g_n^{**}$ converges to $g^*$ under $L_2$-norm. This together with the fact that $I(f,g)$ is continuous in $(f,g)$ implies that there exist some continuous functions $f^{**}$ and $g^{**}$, such that $I(f^{**},g^{**}) \neq 0$.

A key observation here is that, the class of neural networks have universal approximation property. Since the support of $X$ and $Y$ are bounded, it follows from Theorem 1 of [**?** ] that the class of single-layered neural networks with sigmoid activation function is dense in the class of bounded, continuous functions with a compact support. As such, we can find some neural network functions $f^{***}$ and $g^{***}$ such that $I(f^{***},g^{***}) \neq 0$. We then argue that there must exist $f \in \mathbb{H}_1$ and $g \in \mathbb{H}_2$, such that $I(f,g) = 0$. Otherwise, $f^{***}$ and $g^{***}$ can be represented as linear combinations of neural network functions in $\mathbb{H}_1$, $\mathbb{H}_2$ with finitely many number of parameters, and we would have $I(f^{***},g^{***}) = 0$ as a result. This completes Step 1.

**Step 2.** We first show that $I(h_{1,\theta_1}, h_{2,\theta_2})$ is a Lipschitz continuous function of $(\theta_1,\theta_2)$. Note that $h_{1,\theta_1}(X)$ and $h_{2,\theta_2}(Y)$ are Lipschitz continuous functions of $\theta_1$ and $\theta_2$, respectively. For

any $\theta_{1,1}, \theta_{1,2} \in \mathbb{R}^{d_1}$, $\theta_{2,1}, \theta_{2,2} \in \mathbb{R}^{d_2}$, we have that,

$$
\begin{aligned}
&|I(h_{1,\theta_1}, h_{2,\theta_2}) - I(h_{1,\theta_1}, h_{2,\theta_2})| \\
\leq\ &\left|\mathrm{E}[h_{1,1}(X) - \mathrm{E}\{h_{1,1}(X)|Z\} - h_{1,2}(X) + \mathrm{E}\{h_{2,1}(X)|Z\}][h_{2,1}(Y) - \mathrm{E}\{h_{2,1}(Y)|Z\}]\right| \qquad \text{(D.1.17)} \\
&+ \left|\mathrm{E}[h_{1,2}(X) - \mathrm{E}\{h_{1,2}(X)|Z\}][h_{2,1}(Y) - \mathrm{E}\{h_{2,1}(Y)|Z\} - h_{2,2}(Y) + \mathrm{E}\{h_{2,2}(Y)|Z\}]\right|. \qquad \text{(D.1.18)}
\end{aligned}
$$

Since the class of functions in $\mathbb{H}_2$ are upper bounded by $O(\sqrt{\log n})$ with probability tending to 1, the right-hand-side of (D.1.17) is bounded from above by

$$
O(1)\mathrm{E}\left|h_{1,1}(X) - \mathrm{E}\{h_{1,1}(X)|Z\} - h_{1,2}(X) + \mathrm{E}\{h_{2,1}(X)|Z\}\right| \sqrt{\log n},
$$

with probability tending to 1. By Jensen's inequality, the above quantity can be further bounded from above by

$$
O(1)\mathrm{E}\left|h_{1,1}(X) - h_{1,2}(X)\right| 2\sqrt{\log n} \leq K\|\theta_{1,1} - \theta_{1,2}\|_2 \sqrt{\log n},
$$

for some constant $K > 0$. Following similar arguments, we can show that the right-hand-side of (D.1.18) is bounded from above by $K\|\theta_{2,1} - \theta_{2,2}\|_2\sqrt{\log n}$, for any $\theta_{2,1}$ and $\theta_{2,2}$, with probability tending to 1. To summarize, conditional on the event that $\mathcal{H}_1$ and $\mathcal{H}_2$ are bounded function classes, we have shown that

$$
|I(h_{1,\theta_1}, h_{2,\theta_2}) - I(h_{1,\theta_1}, h_{2,\theta_2})| \leq K\left(\|\theta_{1,1} - \theta_{1,2}\|_2 + \|\theta_{2,1} - \theta_{2,2}\|_2\right) \sqrt{\log n}.
$$

Consequently, for any sufficiently small $\varepsilon > 0$, there exists a neighborhood $\mathcal{N} = \{(\theta_1, \theta_2) : \|\theta_j - \theta_j^*\|_2 \leq \delta \log^{-1/2} n\}$ for some constant $\delta > 0$ around $(\theta_1^*, \theta_2^*)$, such that $I(h_{1,\theta_1}, h_{2,\theta_2}) \geq \varepsilon$ for any $(\theta_1, \theta_2)$ that belongs to this neighborhood.

Since $(\theta_{1,b}, \theta_{2,b})$ are generated from the multivariate normal distribution, and the dimensions $d_1$ and $d_2$ are finite, the probability that $(\theta_{1,b}, \theta_{2,b})$ belongs to this neighborhood is strictly greater than $O(\log^{-c_1} n)$ for some constant $c_1 > 0$. Since $B = c_0 n^c$, the probability that at least one pair of parameters $(\theta_{1,b_1}, \theta_{2,b_2})$ belongs to this neighborhood approaches one. Consequently, we have that,

$$
\max_{b_1, b_2} \mathrm{GCM}^*\left\{h_{1,b_1}(X), h_{2,b_2}(Y)\right\} \geq \varepsilon,
$$

with probability tending to 1.

Following similar arguments as in the proof of Theorems 6.4.2 and 6.4.3, we can show that $|T - \max_{b_1,b_2} \text{GCM}^*\{h_{1,b_1}(X), h_{2,b_2}(Y)\}| = o_p(1)$, and $\widetilde{T}_j = o_p(1)$. Consequently, both probabilities $\Pr(T < \varepsilon/2)$ and $\Pr(\widetilde{T}_j \geq \varepsilon/2)$ converge to zero. Therefore, the probability that the $p$-value is greater than $\alpha$ is bounded by the probability that $\Pr(T < \varepsilon/2)$, which converges to zero. This completes the proof of Theorem 6.4.4.