Conditional COT-GAN for Video Prediction with Kernel Smoothing

Anonymous Author(s)
Affiliation
Address
email

Abstract

Causal Optimal Transport (COT) results from imposing a temporal causality constraint on classic optimal transport problems. Relying on recent work of COT-GAN [33] optimized for sequential learning, the contribution of the present paper is twofold. First, we develop a conditional version of COT-GAN suitable for sequence prediction. This means that the dataset is now used in order to learn how a sequence will evolve given the observation of its past evolution. Second, we improve on the convergence results by working with modifications of the empirical measures via kernel smoothing. The resulting *kernel conditional* COT-GAN (KCCOT-GAN) algorithm is illustrated with an application for video prediction.

1 Introduction

Spatio-temporal learning is a challenging task. A desirable model should not only capture the distribution of spatial features at each time step, but also learn its evolution over time. Prior works typically rely on suitable network architectures to capture this complex spatio-temporal structure [26, 1, 24, 30, 27]. At the same time, the recent advances in the field of causal optimal transport (COT) have shown promising developments of loss functions for sequence comparison [4, 5, 21, 33]. This type of transport constrains the transport plans to respect temporal causality in a way that, at every time step, we only use information available up to that time. This provides the foundation of COT-GAN [33], which proved to be an efficient tool that produces high-quality video sequences.

As noted in [22] and [5], causal distances 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 [22] proposed a convoluted empirical measure with a scaled smoothing kernel, while Backhoff et al. [5] 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 paper, we follow the approach of adapting the empirical measure by kernel smoothing, and prove that the resulting adapted empirical measure is also a strongly consistent estimator with respect to COT. Furthermore, we extend the COT-GAN to a conditional framework, to predict how a sequence is likely to evolve given the observation of its past evolution. Finally, we show that our kernel conditional COT-GAN algorithm achieves state-of-the-art results for video prediction.

2 Causal Optimal Transport

Given two probability measures μ, ν defined on $\mathbb{R}^{d\times T}$, $d \times T \in \mathbb{N}$, and a cost function $c : \mathbb{R}^{d\times T} \times \mathbb{R}^{d\times T} \to \mathbb{R}$, the causal optimal transport of μ into ν is formulated as

$$\mathcal{W}_c^{\mathcal{K}}(\mu, \nu) := \inf_{\pi \in \Pi^{\mathcal{K}}(\mu,\nu)} \mathbb{E}^{\pi}[c(x, y)], \tag{1}$$

Submitted to Workshop on Robustness in Sequence Modeling at the 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

where $\Pi^{\mathcal{K}}(\mu,\nu)$ is the set of probability measures on $\mathbb{R}^{d\times T}\times\mathbb{R}^{d\times T}$ with marginals μ,ν, which are called *causal transport plans* between μ and ν if they satisfy the constraint

$$\pi(dy_t|dx_{1:T}) = \pi(dy_t|dx_{1:t}) \qquad \text{for all } t = 1,\cdots,T-1, \tag{2}$$

where $x = (x_1,...,x_T)$ and $y = (y_1,...,y_T)$ are the first and second half of the coordinates on $\mathbb{R}^{d\times T}\times\mathbb{R}^{d\times T}$, and $x_{s:t} = (x_s,...,x_t)$ for all $s < t$. Intuitively, the probability mass moved to the arrival sequence at time t only depends on the starting sequence up to time t.

The *nested distance* [21] or *adapted Wasserstein* distance [5] is the result of an optimal transport problem where plans are required to satisfy the causality constraint as well as its symmetric counterpart, when inverting the role of x and y. Denoting the inverse transport plan $\pi'(dx,dy) = \pi(dy,dx)$, the nested distance [21] or adapted Wasserstein distance is defined as

$$\mathcal{AW}_c(\mu,\nu) := \inf\{\mathbb{E}^{\pi}[c(x,y)] : \pi\in\Pi^{\mathcal{K}}(\mu,\nu), \pi'\in\Pi^{\mathcal{K}}(\nu,\mu)\}. \tag{3}$$

3 Conditional COT-GAN

Consider 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 μ on $\mathbb{R}^{d\times T}$, from which we want to extract other sequences. COT-GAN [33] learns to generate a sample distribution to be similar to the data distribution μ by training a generator g_θ via a min-max objective function equivalent to $\mathcal{W}_c^{\mathcal{K}}$ in (1).

Here, the conditional learning will be done via a conditional generative adversarial structure analogously to [33]. Given a minibatch $\{x_{1:T}^i\}_{i=1}^m$ from the dataset and a sample $\{z_{k+1:T}^i\}^m$ from a noise distribution ζ on some latent space $\mathcal{Z}$, we deploy the generator g_θ, parameterized by θ, to predict the future evolution $\hat{x}_{k+1:T}^i = g_\theta(x_{1:k}^i, z_{k+1:T}^i)$ of $x_{1:k}^i$. The prediction is then concatenated with the corresponding input sequence over the time dimension in order to be compared with the training sequence by the discriminator. We denote the empirical distributions of real and concatenated data by

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

where $\widehat{\nu}_\theta^c$ incorporates the parameterization of g_θ through $\{\hat{x}_{k+1:T}^i\}_{i=1}^m$.

4 Adapted Empirical Measure and KCCOT-GAN

For any measure μ, and for the empirical measures $\widehat{\mu}_m$ relative to a random sample of size m from it, it is known (see e.g. [11]) that Wasserstein distance $\mathcal{W}_c(\mu,\widehat{\mu}_m)\to 0$ as $m\to\infty$, whereas [5] and [22] 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 [33]. Following [22], we obtain the adapted empirical measure via kernel smoothing in order to yield a better convergence guarantee.

For a probability measure μ 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 $\hat{f}(x) = \int k_h(x-y)f(y)dy = f * k_h(x)$ 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 $\widehat{\nu}_\theta^c$:

$$\widehat{\mu}^f := \widehat{\mu} * K^f = \sum_{i=1}^m w_i\delta_{x_{1:T}^i}, \quad \widehat{\nu}_\theta^{c,f} := \widehat{\nu}_\theta^c * K^f = \sum_{i=1}^m w_i\delta_{\text{concat}(x_{1:k}^i,\hat{x}_{k+1:T}^i)}, \tag{4}$$

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.

By [22, Theorems 2 and 4], we have that the adapted Wasserstein distance of the convoluted measures converges, i.e., $P(\mathcal{AW}_c(\widehat{\mu}^f,\widehat{\nu}_\theta^{c,f}) > \varepsilon)\to 0$ as $m\to\infty$. While convergence results in [22] are derived for the adapted Wasserstein distance $\mathcal{AW}_c$, it is easy to see that they extend to $\mathcal{W}_c^{\mathcal{K}}$ since

$$\mathcal{W}_c^{\mathcal{K}}(\mu,\nu) \le \mathcal{AW}_c(\mu,\nu) \tag{5}$$

for any probability measures μ, ν 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. (1) and (3).

Inspired by Genevay et al. [12], we estimate COT problems (1) via their entropy-regularized version $\mathcal{P}^{\mathcal{K}}_{c,\varepsilon}(\mu,\nu) := \inf_{\pi\in\Pi^{\mathcal{K}}(\mu,\nu)}\{\mathbb{E}^{\pi}[c(x,y)] - \varepsilon H(\pi)\}$, for $\varepsilon > 0$, where $H(.)$ is the Shannon entropy. This allows us to use the *Sinkhorn divergence*, based on the Sinkhorn algorithm introduced in [9]. The objective function of the KCCOT-GAN at the level of minibatches is then computed as:

$$\widehat{\mathcal{W}}_{c^{\mathcal{K}}_{\varphi},\varepsilon}(\widehat{\mu}^f, \widehat{\nu}^{c,f}_{\theta}) - \lambda p_{\mathbf{M}_{\varphi_2}}(\widehat{\mu}^f), \tag{6}$$

where the first term is the Sinkhorn divergence relative to the cost $c^{\mathcal{K}}_{\varphi}$ and computed on the convoluted measures in (4), and the second one is a martingale regularization; see Appendix (A.2) for details. We maximize over φ to search for a worst-case distance between the two measures, and minimize over θ to learn a conditional distribution that is as close as possible to the real distribution. Implementation details of KCCOT-GAN can be found in Appendix B.

5 Related Work

Many methods for video prediction relying on variational inference [6] and VAE [17], e.g. SV2P [3], SVP-LP [10], VTA [16], and VRNN [7], have shown promising results. The majority of adversarial models adopted in this domain were trained on the original GAN objective [14] or the Wasserstein GAN objective [2], both of which provide step-wise comparison of sequences. SAVP [18] combined the objective function of the original GAN and VAE to achieve the state of the art performance. Among this line of works, substantial efforts have been devoted to designing specific architectures that tackle the spatio-temporal dependencies, e.g. [30, 24, 27, 8, 19, 29], and training schemes that facilitate learning, e.g. [19, 29, 1]. Whilst some works such as TGAN [24] and VGAN [30] combined a static content generator with a motion generator, others, e.g. [27, 8], designed two discriminators to evaluate the spatial and temporal components separately.

Another important direction of research is the identification of more suitable loss functions. Mathieu et al. [19] explored a loss that measures gradient difference at frame level on top of an adversarial loss trained with a multi-scale architecture. TimeGAN [34] 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 [33] explored a more natural formulation for sequential generation which leads to convincing results.

6 Experiments

We compare **KCCOT-GAN** to **CCOT-GAN** without kernel smoothing as an ablation study, to **SVP-LP** [10], to **SAVP** [18], and to **VRNN** [7], 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 $c(x,y) = \sum_t \|x_t - y_t\|_2^2$. We select the first 15 frames and downsample them to a resolution of 64×64. 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. Network architectures and more training details are given in Appendix B. Samples generated by KCCOT-GAN trained on Moving MNIST dataset are shown in Appendix C.

GQN Mazes. Figure 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 1.

BAIR Push Small. 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 2. It is clearly a very difficult

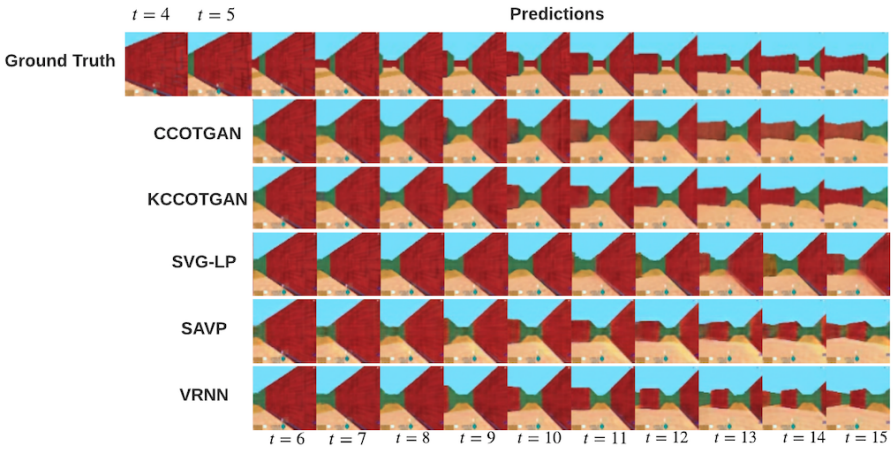

Figure 1: GQN Mazes test results. Only the last 2 frames from the context sequence are shown.

task to outperform these two baselines. On the other hand, SAVP has failed in producing high quality predictions. 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.

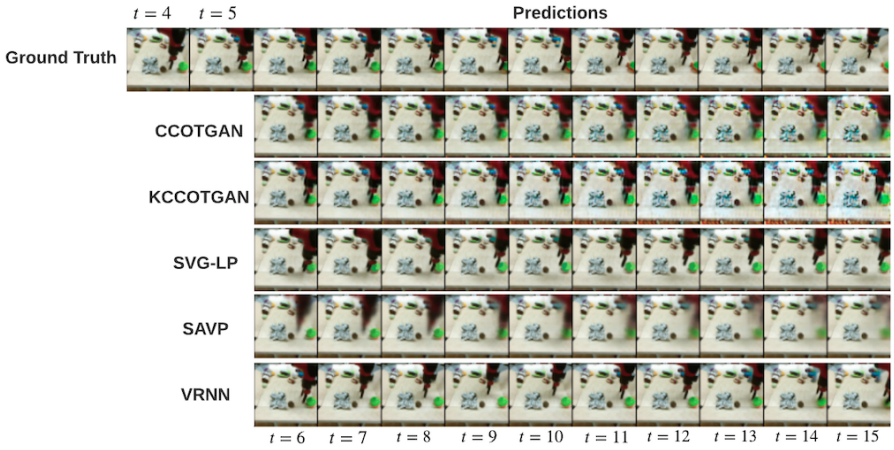

Figure 2: BAIR Push Small test results. Only the last 2 frames from the context sequence are shown.

Evaluation. We evaluate the video predictions using three metrics: Structural Similarity Index [31] (SSIM, higher is better), Learned Perceptual Image Patch Similarity [36] (LPIPS, lower is better), Fréchet Video Distance [28] (FVD, lower is better). The evaluation scores are reported in Table 1.

Table 1: Evaluations for video datasets.

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

References

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

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

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

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

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

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

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

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

[9] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, 2013.

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

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

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

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

[14] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *NIPS*, 2014.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[32] R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[33] T. Xu, L. K. Wenliang, M. Munn, and B. Acciaio. COT-GAN: Generating Sequential Data via Causal Optimal Transport. In *NeurIPS*, 2020.

[34] J. Yoon, D. Jarrett, and M. van der Schaar. Time-series generative adversarial networks. In *NeurIPS*. 2019.

[35] M. Zhang, P. Hayes, T. Bird, R. Habib, and D. Barber. Spread divergence. In *International Conference on Machine Learning*, pages 11106–11116. PMLR, 2020.

[36] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. 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, 2018.

Conditional COT-GAN for Video Prediction with Kernel Smoothing: Supplementary material

A Details on regularized Causal Optimal Transport

A.1 Sinkhorn algorithm

The entropy-regularized transport problems is obtained by considering an entropic constraint. For transport plans with marginals μ supported on a finite set $\{x^i\}_i$ and ν on a finite set $\{y^j\}_j$, any $\pi \in \Pi(\mu, \nu)$ is also discrete with support on the set of all possible pairs $\{(x^i, y^j)\}_{i,j}$. Denoting $\pi_{ij} = \pi(x^i, y^j)$, the Shannon entropy of π is given by $H(\pi) := -\sum_{i,j} \pi_{ij} \log(\pi_{ij})$. A transport plan in the discrete case can be considered as a table identified with a joint distribution. The intuition of imposing such a regularization is to restrict the search of couplings to tables with sufficient smoothness in order to improve efficiency.

When the measures are discrete, such a regularized optimal transport problem becomes easily solvable by using the Sinkhorn algorithm for a given number of iterations, say L, in order to approximate a solution to the Sinkhorn divergence (8), see [12] for detail. Generally speaking, the stronger the regularization is (that is, the bigger the parameter ε is), the fewer number of iterations L are needed in order to yield a good approximation.

A.2 Details about COT-GAN

In this section we will recall the main steps that led to the COT-GAN algorithm for sequential learning in Xu et al. [33], and refer to Appendix A for the details. We then introduce a conditional version, called *conditional* COT-GAN (CCOT-GAN), suited for sequential prediction.

Solving (causal) optimal transport problems is typically computational costly for large datasets. One way to circumvent this challenge is to resort to approximations of transport problems by means of efficiently solvable auxiliary problems. Notably, Genevay et al. [12] proposed the *Sinkhorn divergence*, which allows for the use of the Sinkhorn algorithm [9]. The first observation is that the classical optimal transport

$$\inf_{\pi \in \Pi(\mu,\nu)} \mathbb{E}^{\pi}[c(x, y)]$$

is the limit for $\varepsilon \to 0$ of the entropy-regularized transport problems

$$\mathcal{P}_{c,\varepsilon}(\mu, \nu) := \inf_{\pi \in \Pi(\mu,\nu)} \{\mathbb{E}^{\pi}[c(x, y)] - \varepsilon H(\pi)\}, \quad \varepsilon > 0, \tag{7}$$

where $H(\pi)$ is the Shannon entropy of π. Denoting by $\pi_{c,\varepsilon}(\mu, \nu)$ the optimizer in (7), and by $\mathcal{W}_{c,\varepsilon}(\mu, \nu) := \mathbb{E}^{\pi_{c,\varepsilon}(\mu,\nu)}[c(x, y)]$ the resulting total cost, the Sinkhorn divergence is defined as

$$\widehat{\mathcal{W}}_{c,\varepsilon}(\mu, \nu) := 2\mathcal{W}_{c,\varepsilon}(\mu, \nu) - \mathcal{W}_{c,\varepsilon}(\mu, \mu) - \mathcal{W}_{c,\varepsilon}(\nu, \nu). \tag{8}$$

Similarly, in a causal setting, we consider the entropy-regularized COT problems

$$\mathcal{P}^{\mathcal{K}}_{c,\varepsilon}(\mu, \nu) := \inf_{\pi \in \Pi^{\mathcal{K}}(\mu,\nu)} \{\mathbb{E}^{\pi}[c(x, y)] - \varepsilon H(\pi)\}, \quad \varepsilon > 0, \tag{9}$$

approximating (1).

It is now useful to recall an equivalent characterization of causality: a transport plan $\pi \in \Pi(\mu, \nu)$ is causal if and only if

$$\mathbb{E}^{\pi}\left[\sum_{t=1}^{T-1} h_t(y)\Delta_{t+1}M(x)\right] = 0 \text{ for all } (h, M) \in \mathcal{H}(\mu). \tag{10}$$

With an abuse of notation we write $h_t(y)$, $M_t(x)$, $\Delta_{t+1}M(x)$ rather than $h_t(y_{1:t})$, $M_t(x_{1:t})$, $\Delta_{t+1}M(x_{1:t+1})$. Therefore, the entropy-regularized COT problem (9) can be reformulated as a maximization over regularized transport problems with respect to a specific family of cost functions:

$$\mathcal{P}^{\mathcal{K}}_{c,\varepsilon}(\mu, \nu) = \sup_{c^{\mathcal{K}} \in \mathcal{C}^{\mathcal{K}}(\mu,c)} \mathcal{P}_{c^{\mathcal{K}},\varepsilon}(\mu, \nu). \tag{11}$$

The family of costs $\mathcal{C}^{\mathcal{K}}(\mu, c)$ is given by

$$\mathcal{C}^{\mathcal{K}}(\mu, c) := \left\{ c(x, y) + \sum_{j=1}^{J} \sum_{t=1}^{T-1} h_t^j(y) \Delta_{t+1} M^j(x) : J \in \mathbb{N}, (h^j, M^j) \in \mathcal{H}(\mu) \right\}, \tag{12}$$

where $\Delta_{t+1} M(x) := M_{t+1}(x_{1:t+1}) - M_t(x_{1:t})$ and $\mathcal{H}(\mu)$ is a set of functions depicting causality:

$$\mathcal{H}(\mu) := \{(h, M) : h = (h_t)_{t=1}^{T-1}, h_t \in \mathcal{C}_b(\mathbb{R}^{d \times t}), M = (M_t)_{t=1}^{T} \in \mathcal{M}(\mu),\ M_t \in \mathcal{C}_b(\mathbb{R}^{d \times t})\},$$

with $\mathcal{M}(\mu)$ being the set of martingales on $\mathbb{R}^{d \times T}$ w.r.t. the canonical filtration and the measure μ, and $\mathcal{C}_b(\mathbb{R}^{d \times t})$ the space of continuous, bounded functions on $\mathbb{R}^{d \times t}$. This suggests the following as a robust version of the Sinkhorn divergence from (8) that takes into account causality:

$$\sup_{c^{\mathcal{K}} \in \mathcal{C}^{\mathcal{K}}(\mu, c)} \widehat{\mathcal{W}}_{c^{\mathcal{K}}, \varepsilon}(\mu, \nu).$$

This is the distance used by the discriminator in COT-GAN [33] in order to evaluate the discrepancy between real data and generated one, and it is the one we will use in the current paper for sequential prediction.

Furthermore, [33] makes the two following adjustments needed to make computations feasible. First, rather than considering the whole set of costs in (12), in (11) we optimize over a subset $\mathcal{C}^{\mathcal{K}}(\mu, c)$, by considering $\mathbf{h} := (h^j)_{j=1}^J$ and $\mathbf{M} := (M^j)_{j=1}^J$ of dimension bounded by a fixed $J \in \mathbb{N}$. Second, instead of requiring $\mathbf{M}$ to be a martingale, we consider all continuous bounded functions and introduce a regularization term which penalizes deviations from being a martingale. For a mini-batch of size m, $\{x_{1:T}^i\}_{i=1}^m$, sampled from the dataset, the martingale penalization for $\mathbf{M}$ is defined as

$$p_{\mathbf{M}}(\widehat{\mu}) := \frac{1}{mT} \sum_{j=1}^{J} \sum_{t=1}^{T-1} \left| \sum_{i=1}^{m} \frac{M_{t+1}^j(x_{1:t+1}^i) - M_t^j(x_{1:t}^i)}{\sqrt{\mathrm{Var}[M^j]} + \eta} \right|,$$

where $\widehat{\mu}$ is the empirical measure corresponding to the mini-batch sampled from the dataset, $\mathrm{Var}[M]$ is the empirical variance of M over time and batch, and $\eta > 0$ is a small constant. This leads to the following objective function for COT-GAN in [33]:

$$\widehat{\mathcal{W}}_{c_\varphi^{\mathcal{K}}, \varepsilon}(\widehat{\mu}, \widehat{\nu}_\theta) - \lambda p_{\mathbf{M}_{\varphi_2}}(\widehat{\mu}), \tag{13}$$

where $\widehat{\nu}_\theta$ is the empirical measure corresponding to the mini-batch produced by the generator, parameterized by θ, $\mathbf{h}_{\varphi_1}$ and $\mathbf{M}_{\varphi_2}$ represent the discriminator who learns the worst-case cost $c_\varphi^{\mathcal{K}}$, parameterized by $\varphi := (\varphi_1, \varphi_2)$, and λ is a positive constant.

B Implementation of KCCOT-GAN

B.1 Encoder-decoder structure

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) [25]. The decoder was trained using a hierarchical version of the Teacher Forcing algorithm [32] 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 features of it through an encoder with n layers,

$$\begin{aligned} e_{1:T}^1 &= f_{\theta_e^1}(x_{1:T}), \\ e_{1:T}^2 &= f_{\theta_e^2}(e_{1:T}^1), \\ &\vdots \\ e_{1:T}^n &= f_{\theta_e^n}(e_{1:T}^{n-1}). \end{aligned}$$

Algorithm 1 training KCCOT-GAN by SGD

Input: $\{x^i_{1:T}\}_{i=1}^n$(data), ζ(distribution on latent space)
Parameters: θ_0, φ_0(initialization of parameters), m(batch size), ε(regularization parameter), α(learning rate), λ(martingale penalty coefficient), h(bandwidth parameter)
repeat
(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 ζ;
(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:
$\theta \leftarrow \theta - \alpha \nabla_\theta \left(\widehat{\mathcal{W}}_{c^{\mathcal{K}}_\varphi,\varepsilon}(\widehat{\mu}^f, \widehat{\nu}^{c,f}_\theta) \right)$;
until convergence

From here on, we denote the encoder as f_{θ_e} parametrized by $\theta_e := \{\theta_e^1, \theta_e^2, ..., \theta_e^n\}$, 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_{θ_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 φ are learned on the level of mini-batches via Stochastic Gradient Descent (SGD). The training workflow of KCCOT-GAN is summarized in Algorithm 1.

B.2 Kernel Choice

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. [15, 23, 20, 13], we apply a 3D scaled Gaussian kernel to both spatio and temporal dimensions. In another

Table 2: 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

line of work, Zhang et al. [35] show that convoluting measures with a kernel density estimator is also a valid approach to tackle the problem of disjoint supports in divergence minimization.

Pflug and Pichler [22] proved that the adapted Wasserstein distance of the convoluted measures converges, i.e.,

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

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*

$$h_m \to 0, \quad \frac{mh_m}{|\log h_m|} \to \infty, \quad \frac{|\log h_m|}{\log\log m} \to \infty,$$
$$\textit{and} \quad mh_m \to \infty, \quad \textit{as} \quad m \to \infty, \tag{14}$$

4. *the measures μ and ν are conditionally Lipschitz.*

For proofs and detailed discussions, please see Theorem 2 and 4 in [22]. The result proved for the COT distances in Section 4 is also conditioned on the constraints in Eq. (14).

However, to simplify 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 realize 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.

C Experiment details

C.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.

Table 3: 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

The features from the last encoding layer has a shape of 4×4 (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 2. As the discriminator, the process $\mathbf{h}$ and $\mathbf{M}$ are parameterized with two separate networks that share the same structure, shown in Table 3. 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 η_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 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.

C.2 Results on Moving MNIST

Predictions from KCCOT-GAN conditioned on the first 5 context frames from the test set of the Moving MNIST dataset are presented in Figure 3.

Figure 3: 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.