Sequential Generative Adversarial Networks via Causal Optimal Transport

Beatrice Acciaio

Outline

1. A gentle walk through Generative Adversarial models
2. Our suggestion: Causal Wasserstein GAN
3. Applications
4. Conclusions

Outline

1. A gentle walk through Generative Adversarial models
2. Our suggestion: Causal Wasserstein GAN
3. Applications
4. Conclusions

Generative Adversarial models

Generative: train a **Generator G** to learn data distribution from an i.i.d. sample of observations (training data)

Adversarial: we set a **Discriminator D** against the generator, to stimulate G to do a better job

- In a loop, we train: **G** to generate real-looking samples, and **D** to recognize whether an element comes from real data or is fake (generated by G).
- G and D compete with each other, and the competition drives both of them to improve their performance, until the generated samples are indistinguishable from the genuine data samples (zero-sum game).

$$
\begin{tikzcd}
 & & \square\ \text{real} \arrow[dr] & & \\
 & & & \boxed{\text{D}} \arrow[r] & \square/\lozenge \\
\text{latent} \arrow[r] & \boxed{\text{G}} \arrow[r] & \lozenge\ \text{fake} \arrow[ur] & &
\end{tikzcd}
$$

Generative Adversarial Networks (Goodfellows et al. 2014)

- training data $\{x^i\}_{i=1}^N$ on $\mathcal{X}$, empirical distribution $\mu = \frac{1}{N}\sum_{i=1}^N \delta_{x^i}$
- latent space $\mathcal{Z}$, $\dim(\mathcal{Z}) << \dim(\mathcal{X})$, noise distribution $\zeta \in \mathcal{P}(\mathcal{Z})$
- $g : \mathcal{Z} \to \mathcal{X}$ generates samples, $\nu = g_\# \zeta \in \mathcal{P}(\mathcal{X})$ (cf. μ)
- $f : \mathcal{X} \to [0,1]$ outputs high value if believes input likely to be real

Problem formulation:

$$\inf_g \sup_f \Big\{ \underbrace{\mathbb{E}^{x\sim\mu}[\ln f(x)] + \mathbb{E}^{z\sim\zeta}[\ln(1 - f(g(z)))]}_{\text{objective function}} \Big\}$$

D: learn f s.t. $f(\text{real}) \sim 1$, $f(\text{fake}) \sim 0$

G: learn decoding map g to maximally confuse D

f and g parametrized through Neural Networks $\rightarrow$ f_ϕ, g_θ

Generative Adversarial Networks (Goodfellows et al. 2014)

P: $$\inf_{\theta} \sup_{\phi} \left\{ \mathbb{E}^{x\sim\mu}[\ln f_\phi(x)] + \mathbb{E}^{y\sim\nu_\theta}[\ln(1 - f_\phi(y))] \right\}$$

f_ϕ: parametric family of functions (D's job) $\rightarrow$ NN

$\nu_\theta = g_{\theta\#}\zeta$: parametric family of densities (G's job) $\rightarrow$ NN

$\rightarrow$ **Why not Maximum Likelihood Estimation?**

- Density fitting: $d\nu_\theta(x) = p_\theta(x)dx$
- MLE: $\sup_\theta \frac{1}{N}\sum_{i=1}^N \ln p_\theta(x^i) \longleftrightarrow \inf_\theta \mathrm{KL}(\mu|\nu_\theta)$ (Kullback-Leibler)
- But ν_θ has no density in $\mathcal{X}$, supports of ν_θ and μ may be non-overlapping (MLE not well defined)

$\rightarrow$ If $\{f_\phi\}_\phi, \{g_\theta\}_\theta$ enough capacity, and D trained till optimality:

$$\mathbf{P} \longleftrightarrow \inf_\theta \mathrm{JSD}(\mu|\nu_\theta) \longleftrightarrow \inf_\theta \{\mathrm{KL}(\mu|m) + \mathrm{KL}(\nu_\theta|m)\}$$

(Jensen Shannon Divergence)

Generative Adversarial Networks: moving on

Problems (with original GANs):

- Continuity w.r.t. parameters: $\theta \to \theta' \not\Rightarrow \mathsf{JSD}(\mu|\nu_\theta) \to \mathsf{JSD}(\mu|\nu'_\theta)$
- Convergence: not guaranteed
- Stability: usually unstable

Some ways out:

- Gradient-based regularizations
- Different divergences $\mathfrak{D}(\mu, \nu_\theta)$: Integral Probability Metrics, Maximum Mean Discrepancy, Wasserstein distance, energy distance
- A combination of the above

Example: Wasserstein distance $\mathcal{W}_1(\mu, \nu_\theta) = \inf_{\pi \in \Pi(\mu, \nu_\theta)} \mathbb{E}^\pi[\|x - y\|]$

$$\Longrightarrow \qquad \underbrace{\inf_\theta}_{G} \; \underbrace{\mathcal{W}_1(\mu, \nu_\theta)}_{D}$$

Wasserstein GANs (Arjovsky et al., Gulrajani et al. 2017)

Dual formulation of the Wasserstein distance:

$$\mathcal{W}_1(\mu, \nu_\theta) = \sup_{f \ \mathrm{Lip}_1} \{\mathbb{E}^\mu[f] - \mathbb{E}^{\nu_\theta}[f]\}$$

$\rightarrow$ restrict Kantorovich potentials to have a parametric form f_ϕ

$\rightarrow$ enforce Lip constraint via gradient penalization (easier and regularized)

$$\inf_\theta \sup_\phi \left\{\mathbb{E}^\mu[f_\phi(x)] - \mathbb{E}^{\nu_\theta}[f_\phi(y)] \ + \text{Lip. penalization}\right\}$$

- Continuity: if $\theta \mapsto g_\theta$ cont. $\Rightarrow$ $\theta \mapsto \mathcal{W}_1(\mu, \nu_\theta)$ cont.
- Convergence: WGANs converge if D always trained till optimality
- WGANs outperform MLE and MLE-NN unless exact parametric form of data is known

WGANs $\rightarrow$ Sinkhorn Divergences (Genevay et al. 2017)

Primal problem: numerically more stable (in the dual: gradient requires differentiating dual potential, difficult to compute and unstable)

(i) Consider Wasserstein distance in primal form

(ii) Introduce an entropic penalization to regularize:

$$\mathcal{P}_{c,\varepsilon}(\mu, \nu_\theta) := \inf_{\pi \in \Pi(\mu,\nu_\theta)} \{\mathbb{E}^\pi[c(x,y)] + \varepsilon H(\pi | \mu \otimes \nu_\theta)\} \rightarrow \pi_{c,\varepsilon}(\mu, \nu_\theta)$$

$$\mathcal{W}_{c,\varepsilon}(\mu, \nu_\theta) := \mathbb{E}^{\pi_{c,\varepsilon}(\mu,\nu_\theta)}[c(x,y)]$$

(iii) Learn cost function via parametrization: $c_\phi(x,y) = \|f_\phi(x) - f_\phi(y)\|$

$$\Rightarrow \qquad \inf_\theta \sup_\phi \mathcal{W}_{c_\phi,\epsilon}(\mu, \nu_\theta)$$

▶▶ We will consider a **dynamic framework**: we want to train the generator to generate *discrete-time paths*, given a training set of paths in $\mathcal{X} = \mathbb{R}^{d \times T}$ (or long $\mathbb{R}^d$-valued time series)

Outline

1. A gentle walk through Generative Adversarial models
2. Our suggestion: Causal Wasserstein GAN
3. Applications
4. Conclusions

Causal Wasserstein distance

▶▶ We want a good distance in a dynamic framework

Definition. $\pi \in \mathcal{P}(\mathbb{R}^{d\times T} \times \mathbb{R}^{d\times T})$ is causal if

$$\pi(dy_t|dx_1,\cdots,dx_T) = \pi(dy_t|dx_1,\cdots,dx_t) \quad \forall t$$

$$\Big(\iff \mathbb{E}^\pi\left[\textstyle\sum_{t=1}^{T-1} h_t(y_{\le t})(M_{t+1}(x_{\le t+1}) - M_t(x_{\le t}))\right] = 0 \qquad (*)$$
$$\forall\ (h_t)_t, (M_t)_t : h_t, M_t \in C_b(\mathbb{R}^{d\times t}),\ M \text{ is } (\mathbb{R}^{d\times T}, p_{1\#}\pi)\text{-mart.}\Big)$$

Causal Wasserstein distance:

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

$\Pi^{\text{causal}}(\mu,\nu) = \{\pi \in \mathcal{P}(\mathbb{R}^{d\times T} \times \mathbb{R}^{d\times T}) : \pi \text{ causal, with marginals } \mu,\nu\}$

Entropic regularization (A.-Backhoff-Jia 2019)

Regularized Causal Wasserstein distance:

$$\mathcal{P}^{causal}_{c,\varepsilon}(\mu,\nu_\theta) := \inf_{\pi\in\Pi^{\text{causal}}(\mu,\nu)} \Big\{\mathbb{E}^\pi[c(x,y)] + \epsilon H(\pi|\mu\otimes\nu)\Big\},$$

where $H(\pi|\mu\otimes\nu) = \mathbb{E}^\pi\Big[\log\Big(\frac{d\pi}{d\mu\otimes\nu}\Big)\Big]$.

Thanks to $(*)$,

$$\mathcal{P}^{causal}_{c,\varepsilon}(\mu,\nu_\theta) = \inf_{\pi\in\Pi(\mu,\nu)} \sup_{h,M\,mart} \Bigg\{\mathbb{E}^\pi\Bigg[\underbrace{c(x,y) + \sum_{t=1}^{T-1} h_t(y)\Delta_{t+1}M(x)}_{c_{h,M}}\Bigg] + \epsilon H(\pi)\Bigg\}$$

$$\text{“} = \text{”} \sup_{h,M\,mart} \underbrace{\inf_{\pi\in\Pi(\mu,\nu)} \Big\{\mathbb{E}^\pi[c_{h,M}(x,y)] + \epsilon H(\pi)\Big\}}_{\mathcal{P}_{c_{h,M},\varepsilon}(\mu,\nu_\theta)}$$

Causal Wasserstein GAN

- Parametrize $\to$ h_{ϕ_1}, M_{ϕ_2}, and set $\phi = (\phi_1, \phi_2)$, $c_\phi := c_{h_{\phi_1}, M_{\phi_2}}$
- Eliminate entropic bias $\mathcal{W}_{c_\phi,\epsilon}(\mu,\mu) \neq 0 \to$ consider Sinkhorn loss:

$$\widehat{\mathcal{W}}_{c_\phi,\epsilon}(\mu,\nu) := \mathcal{W}_{c_\phi,\epsilon}(\mu,\nu) - \frac{1}{2}\mathcal{W}_{c_\phi,\epsilon}(\mu,\mu) - \frac{1}{2}\mathcal{W}_{c_\phi,\epsilon}(\nu,\nu)$$

Causal Wasserstein GAN:

$$\boxed{\inf_\theta \sup_\phi \widehat{\mathcal{W}}_{c_\phi,\epsilon}(\mu,\nu_\theta)}$$

  - c_ϕ learned by D through a Recurrent-NN
  - $\nu_\theta = g_{\theta\#}\zeta$, where g_θ learned by G through a Recurrent-NN

- Here $\neq$ Genevay et al: $\mathcal{W}^{causal}$ vs $\mathcal{W}$, RNNs vs NNs

The algorithm

To solve the min-max problem, we approximate $\mathcal{W}_{c_\phi,\epsilon}(\mu, \nu_\theta)$:

(1) sampling mini-batches

(2) penalizing M non-martingale

(3) taking a pre-determined n. of iterations in the Sinkhorn algorithm

(1): Sample mini-batch $\{x^i\}_{i=1}^m$ from the dataset, and sample $\{z^i\}_{i=1}^m$ from the latent space and set $y_\theta^i = g_\theta(z^i)$. Empirical measures:

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

(2): Penalize M_{ϕ_2} non-martingale via $\lambda p_\phi(\hat{\mathbf{x}}^m)$, with $\lambda > 0$ and

$$p_\phi(\hat{\mathbf{x}}^m) := \frac{1}{m}\sum_{t=1}^{T-1}\left|\sum_{i=1}^m \Delta M_{\phi_2,t+1}(x^i)\right|.$$

The algorithm

(3): Compute $\inf_{\pi \in \Pi(\hat{\mathbf{x}}^m, \hat{\mathbf{y}}^m_\theta)} \left\{ \mathbb{E}^\pi[c_\phi] + \epsilon H(\pi | \hat{\mathbf{x}}^m \otimes \hat{\mathbf{y}}^m_\theta) \right\}$

by Sinkhorn algorithm (Cuturi 2013): fast and stable matrix scaling algorithm (via Sinkhorn's fixed point iteration), converges to the unique solution $\pi^* = \mathrm{diag}(u) \exp^{-c_\phi/\epsilon} \mathrm{diag}(v)$.

After L iterations: $\mathcal{W}^{(L)}_{c_\phi,\epsilon}(\hat{\mathbf{x}}^m, \hat{\mathbf{y}}^m_\theta)$, smooth proxy that can be differentiated in a fast and stable way

→ (1)+(2)+(3) $\Rightarrow$ the objective function is:

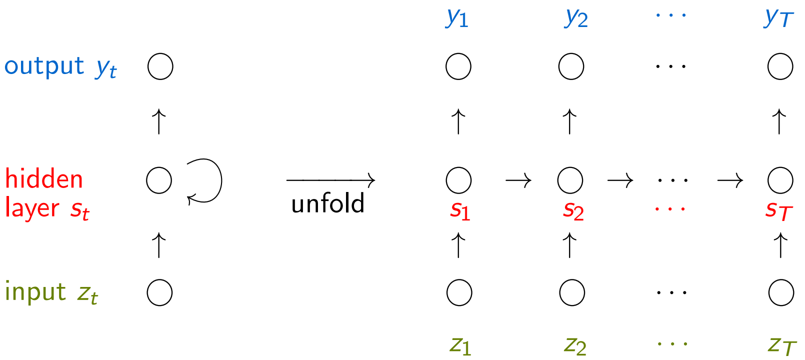
$$V := \widehat{\mathcal{W}}^{(L)}_{c_\phi,\epsilon}(\hat{\mathbf{x}}^m, \hat{\mathbf{y}}^m_\theta) - \lambda p_\phi(\hat{\mathbf{x}}^m)$$

→ Stochastic Gradient Ascent/Descent to update parameters:

$$\phi_{n+1} = \phi_n + \text{“}\alpha \nabla_\phi V\text{”}$$
$$\theta_{n+1} = \theta_n - \text{“}\alpha \nabla_\theta V\text{”}$$

Training architecture

(Basic) Recurrent Neural Network (for G)



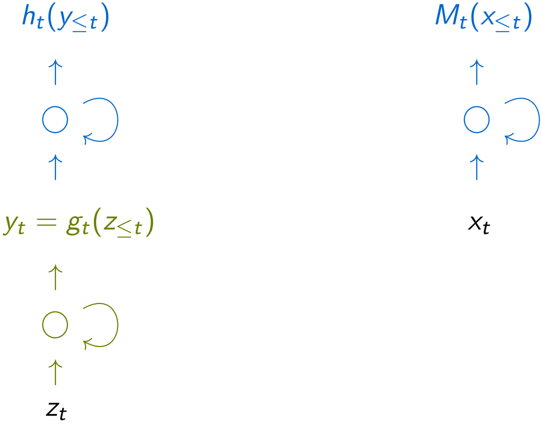
$s_t = \sigma(Az_t + Bs_{t-1} + a)$ history embedding vector (network memory)

$y_t = Cs_t$, σ activation functions, applied component-wise

$\theta = \{A, B, C, a\}$ parameters: weight matrices and bias vectors

Training architecture

Recurrent Neural Networks: G and D



Many alternatives: number of layers, mix with fully connected layers, Long Short Term Memory, Gated Recurrent Unit,...

Pseudo-code

Data: θ_0, ϕ_0, $\{x^i\}_{i=1}^N$(real data), ϵ (entr. coeff.), m (batch size), L (Sinkhorn iterations), α (learning rate), n_c (critic iterations), λ (martingale coeff.)

Result: θ, ϕ

$\theta \leftarrow \theta_0$, $\phi \leftarrow \phi_0$

for $k = 1, 2, \ldots$ **do**

  for $l = 1, 2, \ldots, n_c$ **do**

    Sample: $\{x^i\}_{i=1}^m$ from real data, and $\{z^i\}_{i=1}^m$ from ζ

    $y^i \leftarrow g_\theta(z^i)$

    $\nabla_\phi V \leftarrow \mathtt{AutoDiff}_\phi\Big(\widehat{\mathcal{W}}^{(L)}_{c_\phi,\epsilon}(\hat{\mathbf{x}}^m, \hat{\mathbf{y}}^m_\theta) - \lambda p_\phi(\hat{\mathbf{x}}^m)\Big)$

    $\phi \leftarrow \phi + \alpha \mathtt{RMSProp}(\nabla_\phi V)$

  end

  Sample: $\{x^i\}_{i=1}^m$ from real data, and $\{z^i\}_{i=1}^m$ from ζ

  $y^i \leftarrow g_\theta(z^i)$

  $\nabla_\theta V \leftarrow \mathtt{AutoDiff}_\theta\Big(\widehat{\mathcal{W}}^{(L)}_{c_\phi,\epsilon}(\hat{\mathbf{x}}^m, \hat{\mathbf{y}}^m_\theta)\Big)$

  $\theta \leftarrow \theta - \alpha \mathtt{RMSProp}(\nabla_\theta V)$

end

Looking forward

→ We have been testing some easy-to check features on simulated data, e.g. reproducing periodic curves.

→ Now we start testing on reference databases and real data:

- static: MNIST
- dynamic: music

→ Next main step: develop a **conditional modification of the algorithm**, so that we feed the beginning of a sequence and the generator produces the rest:

- Mathematically: easy modification
- But may require different tuning

Outline

1. A gentle walk through Generative Adversarial models
2. Our suggestion: Causal Wasserstein GAN
3. Applications
4. Conclusions

Applications

$\rightarrow$ Original motivation of CWGANs: learn how to generate real-looking evolutions, given an observed dataset. E.g.

- Natural language processing: text generation.
- Text to speech conversion systems.
- Financial perspective: application to obtain model-independent pricing of financial derivatives.

$\rightarrow$ Depending on the datasets are we interested in, and the features of the evolution we want to capture, architecture and parameters will need to be chosen/tuned.

$\rightarrow$ We will see now: use of it to study Cournot-Nash equilibria

Cournot-Nash equilibrium (A.-Backhoff 2019)

Setting:

- Discrete time $t = 1, ..., T$; game played at time $t = 1$
- N agents whose types x evolve in time: $\mathcal{X}$ path-space of types
- $\mu \in \mathcal{P}(\mathcal{X})$: agents' types distribution
- agents select non-anticipative actions y in time: $\mathcal{Y}$ path-space of actions
- agents face a cost $F(x, y, \nu)$ that depends on their own type, action, and on the mean-field interaction with the rest of the population

Problem:

find **Nash equilibria** (for large systems of players, approximate this problem with asymptotic problem for a representative agent)

Cournot-Nash equilibrium

Cost function $F(x, y, \nu) : \mathcal{X} \times \mathcal{Y} \times \mathcal{P}(\mathcal{Y}) \to \mathbb{R}$

Definition

$\pi^* \in \Pi^{\text{causal}}(\mu, .)$ is called Cournot-Nash equilibrium if:

π^* attains $\inf_{\pi \in \Pi^{\text{causal}}(\mu,.)} \mathbb{E}^{\pi}[F(x, y, \nu^*)]$, and $p_{2\#}\pi^* = \nu^*$

The above is the correct asymptotic formulation of the N-agent problem, in the following sense:

Theorem (A.-Backhoff 2019)

Under some regularity conditions,

1. *CN equilibria provides ϵ-Nash equilibria for N-player game*
2. *when Nash equilibria converge, the limits are CN equilibria*

Cournot-Nash equilibrium: reformulation

Separable cost: $F(x, y, \nu) = f(x, y) \; + \; \underbrace{V[\nu](y)}_{\text{mean-field interaction}}$

Potential game: V first variation of $\mathcal{E}$, $\mathcal{E} : \mathcal{P}(\mathcal{Y}) \to \mathbb{R}$ convex,

$$\lim_{\epsilon \to 0^+} \frac{\mathcal{E}(\nu + \epsilon(\xi - \nu)) - \mathcal{E}(\nu)}{\epsilon} = \int_{\mathcal{Y}} V[\nu] d(\xi - \nu)$$

Theorem (A.-Backhoff 2019)

The following are equivalent:

(i) π^* *is a Cournot-Nash equilibrium;*

(ii) $(p_{2\#}\pi^*, \pi^*)$ *solves the variational problem:*

$$(VP) \qquad \inf_{\nu \in \mathcal{P}(\mathcal{Y})} \left\{ \mathcal{W}_f^{causal}(\mu, \nu) + \mathcal{E}[\nu] \right\}$$

Cournot-Nash equilibrium via CWGANs

Causal Wasserstein GAN:

$$\inf_{\nu} \mathcal{W}_c^{\text{causal}}(\mu,\nu) \quad \hookrightarrow \quad \inf_{\theta} \sup_{\phi} \widehat{\mathcal{W}}_{c_\phi,\epsilon}(\mu, g_{\theta\#}\zeta)$$

$\rightarrow$ we parametrized the set of decoding maps: $g_\theta \rightarrow \nu_\theta = g_{\theta\#}\zeta$

$\rightarrow$ we parametrized the causality constraint: learn cost c_ϕ

$\rightarrow$ we regularized via entropic penalization and corrected the bias

Variational problem ($\sim$ CN equilibria):

$$\inf_{\nu\in\mathcal{P}(\mathcal{Y})} \left\{ \mathcal{W}_f^{\text{causal}}(\mu,\nu) + \mathcal{E}[\nu] \right\} \hookrightarrow \inf_{\theta} \sup_{\phi} \left\{ \widehat{\mathcal{W}}_{f_\phi,\epsilon}(\mu, g_{\theta\#}\mu) + \mathcal{E}[g_{\theta\#}\mu] \right\}$$

Conceptual difference:

$\rightarrow$ we parametrize the transport maps g_θ that push forward the type μ into the action ν. How restrictive is this?

Cournot-Nash equilibrium via CWGANs

→ With the CWGAN approach: we are restricting attention to pure-equilibria distributions: $\nu_\theta = g_{\theta\#}\mu$, with g_θ modelled by an RNN

- Note that
$$(VP) = \inf_{\Pi^{\text{causal}}(\mu,.)} \{\mathbb{E}^\pi[f] + \mathcal{E}(p_{2\#}\pi)\},$$
  and recall that Monge causal transports (pure adapted equilibria) are dense in the set of Kantorovich transports (mixed non-anticipative equilibria): $\overline{\Pi^{\text{adapt.}}(\mu,.)}^w = \Pi^{\text{causal}}(\mu,.)$ (Lacker 2018)
- Basic RNNs are universal approximators of open dynamical systems (Schäfer-Zimmermann 2007):
$$\begin{cases} s_t = \varphi_2(s_{t-1}, z_t) \\ y_t = \varphi_1(s_t) \end{cases}$$
  as long as activation functions σ_i increasing, bounded and continuous

→ We shall compare with numerics in A.-Backhoff-Jia 2019

Outline

1. A gentle walk through Generative Adversarial models
2. Our suggestion: Causal Wasserstein GAN
3. Applications
4. Conclusions

Conclusions

Presented today

- Suggestion of a new dynamic generative adversarial model, through Causal Wasserstein distance and RNN architecture
- Some initial testing
- Possible application to study Cournot-Nash equilibria

To-do list

- Test on real data, tune parameters accordingly, explore different RNN structures (depths, activation functions...)
- Compare with ‘static’ WGANs treating paths as static objects
- Extend to conditional CWGANs, to predict the evolution of an observed path

Literature

- Acciaio, Backhoff: Nash equilibria and OT in a dynamic setting, 2019
- Acciaio, Backhoff, Jia: Numerical computation of COT, 2019
- Arjovsky, Chintala, Bottou: Wasserstein GAN, 2017
- Cuturi: Sinkhorn distances: Lightspeed computation of OT, 2013
- Genevay, Peyré, Cuturi: Learning Generative Models with Sinkhorn Divergences, 2017
- Goodfellows et al.: Generative Adversarial Networks, 2014
- Gulrajani et al.: Improved Training of Wasserstein GANs, 2017
- Lacker: Dense sets of joint distributions appearing in filtration enlargements, stochastic control, and causal optimal transport, 2018
- Schäfer, Zimmermann: RNNs are universal approximators, 2007

Thank you for your attention!