

# A deep learning strategy for solving physics-based Bayesian inference problems

Deep Ray  
Department of Mathematics  
University of Maryland, College Park

Email: [deepray@umd.edu](mailto:deepray@umd.edu)  
Website: [deepray.github.io](http://deepray.github.io)

Supported by ARO grant W911NF2010050 and CARG-USC.

CMAI Colloquium  
February 16, 2024



- ▶ Challenges with Bayesian inference
- ▶ Conditional Wasserstein generative adversarial networks (cWGANs)
- ▶ Learning posteriors with cWGANs
- ▶ Theoretical issues and a new formulation
- ▶ Numerical results
- ▶ Conclusion

## Consider a forward problem

$$\mathcal{F} : \mathbf{x} \in \Omega_x \mapsto \mathbf{y} \in \Omega_y, \quad \Omega_x \in \mathbb{R}^{N_x}, \quad \Omega_y \in \mathbb{R}^{N_y}$$

For example the **heat conduction** PDE:

$$\frac{\partial u(\mathbf{s}, t)}{\partial t} - \kappa \Delta u(\mathbf{s}, t) = 0$$

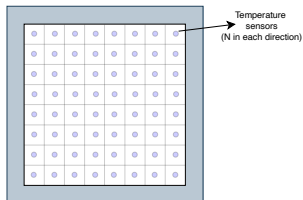
where

$$u(\mathbf{s}, 0) = u_0(\mathbf{s})$$

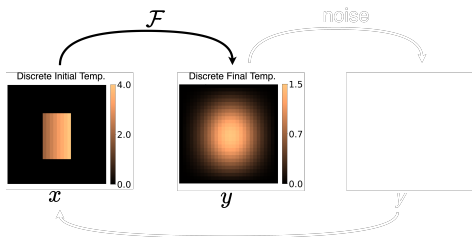
$u(\mathbf{s}, t)$  → temperature at location  $\mathbf{s}$  at time  $t$

$u_0(\mathbf{s})$  → initial temperature at location  $\mathbf{s}$

$\kappa$  → thermal conductivity of material



Forward problem  $\mathcal{F}$ : Given  $u_0(\mathbf{s})$  at the sensor nodes determine  $u(\mathbf{s}, T)$



## Consider a forward problem

$$\mathcal{F} : \mathbf{x} \in \Omega_x \mapsto \mathbf{y} \in \Omega_y, \quad \Omega_x \in \mathbb{R}^{N_x}, \quad \Omega_y \in \mathbb{R}^{N_y}$$

For example the **heat conduction** PDE:

$$\frac{\partial u(\mathbf{s}, t)}{\partial t} - \kappa \Delta u(\mathbf{s}, t) = 0$$

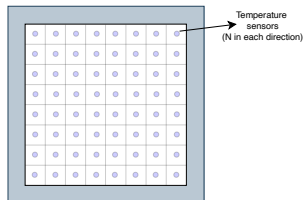
where

$$u(\mathbf{s}, 0) = u_0(\mathbf{s})$$

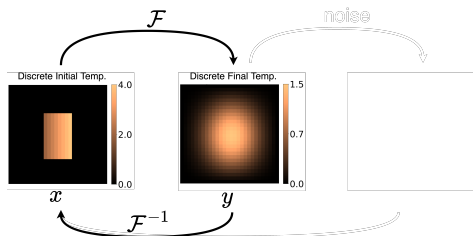
$u(\mathbf{s}, t)$  → temperature at location  $\mathbf{s}$  at time  $t$

$u_0(\mathbf{s})$  → initial temperature at location  $\mathbf{s}$

$\kappa$  → thermal conductivity of material



Inverse problem  $\mathcal{F}^{-1}$ : Given  $u(\mathbf{s}, T)$  at the sensor nodes infer  $u_0(\mathbf{s})$



## Consider a forward problem

$$\mathcal{F} : \mathbf{x} \in \Omega_x \mapsto \mathbf{y} \in \Omega_y, \quad \Omega_x \in \mathbb{R}^{N_x}, \quad \Omega_y \in \mathbb{R}^{N_y}$$

For example the **heat conduction** PDE:

$$\frac{\partial u(\mathbf{s}, t)}{\partial t} - \kappa \Delta u(\mathbf{s}, t) = 0$$

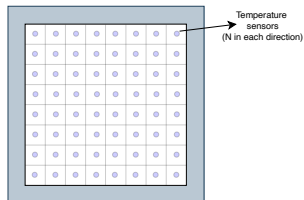
where

$$u(\mathbf{s}, 0) = u_0(\mathbf{s})$$

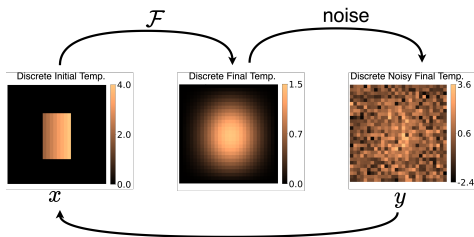
$u(\mathbf{s}, t)$  → temperature at location  $\mathbf{s}$  at time  $t$

$u_0(\mathbf{s})$  → initial temperature at location  $\mathbf{s}$

$\kappa$  → thermal conductivity of material



Inverse problem  $\mathcal{F}^{-1}$ : Given **noisy**  $u(\mathbf{s}, T)$  infer  $u_0(\mathbf{s})$



Challenges with inverse problems:

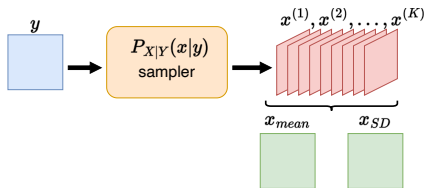
- ▶ Inverse map is **not well posed**.
- ▶ **Noisy measurements**.
- ▶ Need to encode **prior knowledge** about  $x$ .

**Bayesian framework:**  $x$  and  $y$  modelled by random variables  $X$  and  $Y$ .

**AIM:** Given a measurement  $Y = y$  approximate the conditional (posterior) distribution

$$P_{X|Y}(x|y)$$

and sample from it.



- ▶ Posterior sampling techniques, such as Markov Chain Monte Carlo, are prohibitively expensive when **dimension of  $X$  is large**.
- ▶ Characterization of **priors for complex data**

For example,  $x$  data might look like:

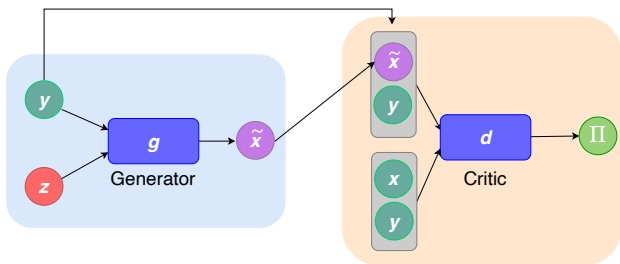


Representing this data using simple distributions is **hard!**

**Resolve both issues using deep learning**

# Conditional Generative Adversarial Network (cGAN)

- ▶ Proposed by Mirza et al. (2014)
- ▶ Learning distributions conditioned on another field.
- ▶ Comprises two neural networks,  $g$  and  $d$ .



Generator network:

- ▶  $g : \Omega_z \times \Omega_y \rightarrow \Omega_x$ .
- ▶ Latent variable  $\mathbf{Z} \sim P_Z$ , e.g.  $N(0, \mathbf{I})$ . Also  $N_z \ll N_x$ .
- ▶  $(\mathbf{x}, \mathbf{y})$  sampled from true  $P_{XY}$

Critic network:

- ▶  $d : \Omega_x \times \Omega_y \rightarrow \mathbb{R}$ .
- ▶  $d$  tries to detect fake samples.
- ▶  $d(\mathbf{x}, \mathbf{y})$  large for real  $\mathbf{x}$ , small otherwise.



## Conditional Wasserstein GAN (cWGAN)

- ▶ A cGAN variant proposed by Adler et al. (2018).
- ▶ Given  $Y = \mathbf{y}$  and  $Z \sim P_Z$  we get a random variable

$$X^g = g(Z, \mathbf{y}), \quad X^g \sim P_{X|Y}^g.$$

- ▶ A cGAN variant proposed by Adler et al. (2018).
- ▶ Given  $Y = y$  and  $Z \sim P_Z$  we get a random variable

$$X^g = g(Z, y), \quad X^g \sim P_{X|Y}^g.$$

- ▶ Objective function

$$\Pi(g, d) = \mathbb{E}_{\substack{(X, Y) \sim P_{XY} \\ Z \sim P_Z}} [d(X, Y) - d(g(Z, Y), Y)]$$

- ▶ Define

$$\text{Lip}_X = \{f : \Omega_x \times \Omega_y \rightarrow \mathbb{R} \text{ s.t. } f \text{ is 1-Lipschitz in } x\}$$

- ▶ A cGAN variant proposed by Adler et al. (2018).
- ▶ Given  $Y = \mathbf{y}$  and  $Z \sim P_Z$  we get a random variable

$$\mathbf{X}^g = g(\mathbf{Z}, \mathbf{y}), \quad \mathbf{X}^g \sim P_{\mathbf{X}^g | \mathbf{Y}}.$$

- ▶ Objective function

$$\Pi(\mathbf{g}, d) = \mathbb{E}_{\substack{(\mathbf{X}, \mathbf{Y}) \sim P_{\mathbf{X}\mathbf{Y}} \\ \mathbf{Z} \sim P_Z}} [d(\mathbf{X}, \mathbf{Y}) - d(\mathbf{g}(\mathbf{Z}, \mathbf{Y}), \mathbf{Y})]$$

- ▶ Define

$$\text{Lip}_X = \{f : \Omega_x \times \Omega_y \rightarrow \mathbb{R} \text{ s.t. } f \text{ is 1-Lipschitz in } \mathbf{x}\}$$

- ▶ Find  $\mathbf{g}^*$  and  $d^*$  by solving the minmax problem

$$d^*(\mathbf{g}) = \arg \max_{d \in \text{Lip}_X} \Pi(\mathbf{g}, d)$$
$$\mathbf{g}^* = \arg \min_{\mathbf{g}} \Pi(\mathbf{g}, d^*(\mathbf{g})).$$

- ▶ Adler et al. (2018) proved that the minmax problem is equivalent to

$$\mathbf{g}^* = \arg \min_{\mathbf{g}} \mathbb{E}_{\mathbf{Y} \sim P_{\mathbf{Y}}} \left[ W_1(P_{\mathbf{X}|\mathbf{Y}}, P_{\mathbf{X}|\mathbf{Y}}^{\mathbf{g}}) \right]$$

where  $W_1$  is the Wasserstein-1 distance (hence the name of the method.)

- ▶  $d^*$  helps estimate the  $W_1$  distance (Kantorovich-Rubinstein duality)

- ▶ Adler et al. (2018) proved that the minmax problem is equivalent to

$$\mathbf{g}^* = \arg \min_{\mathbf{g}} \mathbb{E}_{\mathbf{Y} \sim P_{\mathbf{Y}}} \left[ W_1(P_{\mathbf{X}|\mathbf{Y}}, P_{\mathbf{X}|\mathbf{Y}}^{\mathbf{g}}) \right]$$

where  $W_1$  is the Wasserstein-1 distance (hence the name of the method.)

- ▶  $d^*$  helps estimate the  $W_1$  distance (Kantorovich-Rubinstein duality)
- ▶ How is  $d \in \text{Lip}_X$  enforced?  $\rightarrow$  using a gradient penalty term

$$d^*(\mathbf{g}) = \arg \max_d [\Pi(\mathbf{g}, d) - \lambda \mathcal{GP}_x]$$

where

$$\mathcal{GP}_x = \mathbb{E}_{\delta \sim \mathcal{U}(0,1)} [(\|\partial_1 d(\mathbf{h}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \delta), \mathbf{y})\|_2 - 1)^2]$$

$$\mathbf{h}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \delta) = \delta \mathbf{x} + (1 - \delta) \mathbf{g}(\mathbf{z}, \mathbf{y}).$$

Note the constraint is only on the first argument of  $d$ !

# Posterior sampling using cWGANs

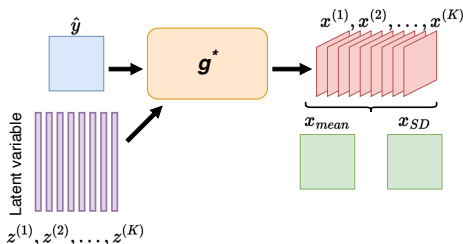
Steps:

- ▶ Acquire samples  $\mathcal{S}_x = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ , where  $\mathbf{x}_i \sim P_X^{\text{prior}}$ .
- ▶ Use forward map  $\mathcal{F}$  to generate paired dataset

$$\mathcal{S} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_N, \mathbf{y}_N)\} \quad \text{where} \quad \mathbf{y}_n = \mathcal{F}(\mathbf{x}_n) + \text{noise}.$$

- ▶ Train a cWGAN on  $\mathcal{S}$ .
- ▶ For a new test measurement  $\hat{\mathbf{y}}$ , generate samples using  $g^*$ .
- ▶ Evaluate statistics using Monte Carlo.

$$\mathbf{X} \sim \mathbb{E}_{P_{\mathbf{X}|\mathbf{Y}}} [\ell(\mathbf{X})] \approx \frac{1}{K} \sum_{i=1}^K \ell(g^*(z^{(i)}, \hat{\mathbf{y}})), \quad z^{(i)} \sim P_Z$$

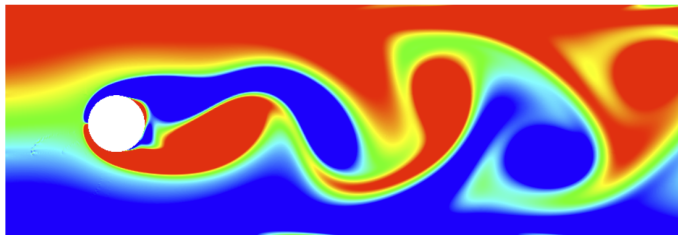


## Interpreting fields as images for some applications

- ▶ RGB image:  $\mathbf{y} \in \mathbb{R}^{N \times M \times C}$ ,  $N \times M \rightarrow$  resolution,  $C = 3 \rightarrow$  channels



- ▶ Grayscale:  $\mathbf{y} \in \mathbb{R}^{N \times M \times 1}$  with a single channel
- ▶ Discrete solution to a PDE in 2D with  $C$  variables:  $\mathbf{y} \in \mathbb{R}^{N \times M \times C}$



Original model by Adler et al.

- ▶ Designed for CT imaging applications
- ▶ In  $g$ , latent variable  $z$  stacked as additional channel of  $y$

$$\text{Input to } g = [y, z] \in \mathbb{R}^{N \times M \times (C+1)}, \quad y \in \mathbb{R}^{N \times M \times C}, \quad z \in \mathbb{R}^{N \times M \times 1}$$

→  $N_z$  scales with  $N_y!$  No dim. reduction

- ▶ Specialized architecture required for critic to avoid mode collapse  
→ larger network and higher training cost!



## Modified version\*

- ▶ Developed and tested on physics-based problems governed by PDEs
- ▶ Conditional instance normalization (CIN) to handle  $g$  inputs of different shapes

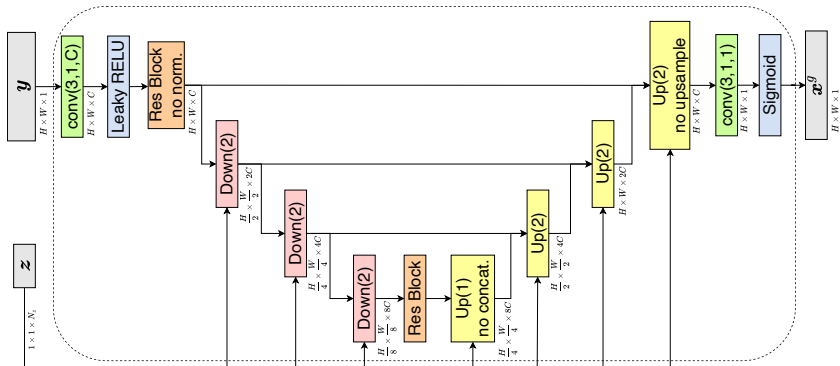
$$\mathbf{y} \in \mathbb{R}^{N \times M \times C}, \quad \mathbf{z} \in \mathbb{R}^{N_z}$$

- $N_z$  no longer depends on  $N_y$  – get dim. reduction!
- introduce multi-level stochasticity (see next slide)

- ▶ Simpler for critic architecture used

\* *The efficacy and generalizability of conditional GANs for posterior inference in physics-based inverse problems* (R., Patel, Ramaswamy, Oberai); Numerical Algebra, Control and Optimization, 2022.

U-Net architecture when  $x, y$  have tensorred (image-like) structure.



Consider the PDE

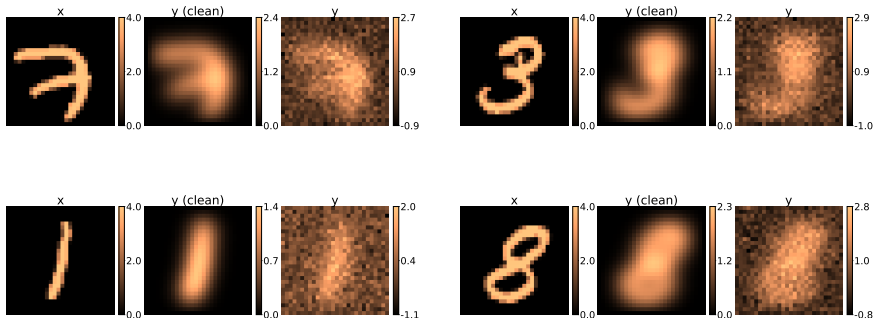
$$\begin{aligned}\frac{\partial u(\mathbf{s}, t)}{\partial t} - \nabla \cdot (\kappa(\mathbf{s}) \nabla u(\mathbf{s}, t)) &= 0, & \forall (\mathbf{s}, t) \in (0, 2\pi)^2 \times (0, 1] \\ u(\mathbf{s}, 0) &= u_0(\mathbf{s}), & \forall \mathbf{s} \in (0, 2\pi)^2 \\ u(\mathbf{s}, t) &= 0, & \forall \mathbf{s} \in \partial(0, 2\pi)^2 \times (0, 1]\end{aligned}$$

- ▶  $x$ : discrete initial temperature field.
- ▶  $y$ : noisy discrete final temperature field.
- ▶  $\mathcal{F}$ : Finite difference solver for the PDE.
- ▶ We assume a constant conductivity  $\kappa = 0.2$ .

Assuming  $x$  to be given by (heat stamped) MNIST handwritten digits and

$$N_x = N_y = 28 \times 28 = 784$$

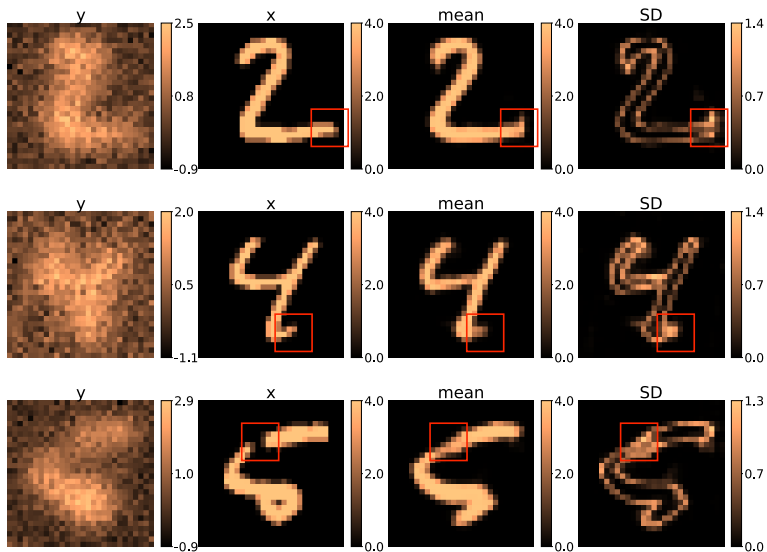
Training samples:



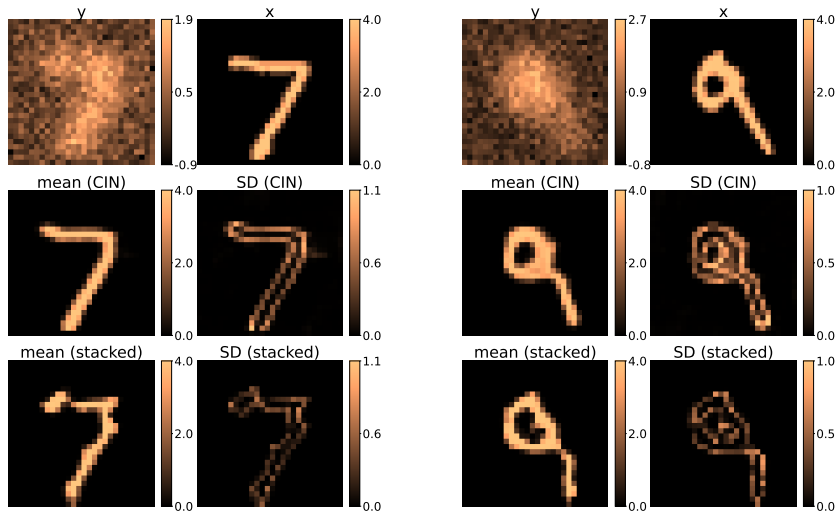
cWGAN trained using latent dimension  $N_z = 100$ .

# SD can be used as a metric for UQ

Testing trained cGAN (statistics with  $K = 800$   $z$  samples)



Low ensemble variability and poor reconstruction with original stacked approach!



See paper for more experiments and a discussion on generalizability!

Recall that

- ▶ We require  $d : \Omega_x \times \Omega_y \rightarrow \mathbb{R}$  to satisfy  $d \in \text{Lip}_x$
- ▶ If the model is “perfectly trained” we have

$$\mathbf{g}^* = \arg \min_{\mathbf{g}} \mathbb{E}_{\mathbf{Y} \sim P_{\mathbf{Y}}} \left[ W_1(P_{\mathbf{X}|\mathbf{Y}}, P_{\mathbf{X}|\mathbf{Y}}^{\mathbf{g}}) \right]$$

- ▶ Original proof by Adler et al. use a **technical assumption** to interchange  $\arg \max_d$  and  $\mathbb{E}_{\mathbf{Y} \sim P_{\mathbf{Y}}}$ , which is not easy to interpret.

Recall that

- ▶ We require  $d : \Omega_x \times \Omega_y \rightarrow \mathbb{R}$  to satisfy  $d \in \text{Lip}_x$
- ▶ If the model is “perfectly trained” we have

$$\mathbf{g}^* = \arg \min_{\mathbf{g}} \mathbb{E}_{\mathbf{Y} \sim P_{\mathbf{Y}}} \left[ W_1(P_{\mathbf{X}|\mathbf{Y}}, P_{\mathbf{X}|\mathbf{Y}}^{\mathbf{g}}) \right]$$

- ▶ Original proof by Adler et al. use a **technical assumption** to interchange  $\arg \max_d$  and  $\mathbb{E}_{\mathbf{Y} \sim P_{\mathbf{Y}}}$ , which is not easy to interpret.

Assume we can find a the sequence  $\{(\mathbf{g}_n^*, d_n^*)\}_n$ , where  $n \in \mathbb{Z}_+$  is the number of weights/biases, s.t.

$$\lim_{n \rightarrow \infty} \Pi(\mathbf{g}_n^*, d_n^*) = \lim_{n \rightarrow \infty} \mathbb{E}_{\mathbf{Y} \sim P_{\mathbf{Y}}} \left[ W_1(P_{\mathbf{X}|\mathbf{Y}}, P_{\mathbf{X}|\mathbf{Y}}^{\mathbf{g}_n^*}) \right] = 0.$$



Recall that

- ▶ We require  $d : \Omega_x \times \Omega_y \rightarrow \mathbb{R}$  to satisfy  $d \in \text{Lip}_x$
- ▶ If the model is “perfectly trained” we have

$$\mathbf{g}^* = \arg \min_{\mathbf{g}} \mathbb{E}_{\mathbf{Y} \sim P_{\mathbf{Y}}} \left[ W_1(P_{\mathbf{X}|\mathbf{Y}}, P_{\mathbf{X}|\mathbf{Y}}^{\mathbf{g}}) \right]$$

- ▶ Original proof by Adler et al. use a **technical assumption** to interchange  $\arg \max_d$  and  $\mathbb{E}_{\mathbf{Y} \sim P_{\mathbf{Y}}}$ , which is not easy to interpret.

Assume we can find a the sequence  $\{(\mathbf{g}_n^*, d_n^*)\}_n$ , where  $n \in \mathbb{Z}_+$  is the number of weights/biases, s.t.

$$\lim_{n \rightarrow \infty} \Pi(\mathbf{g}_n^*, d_n^*) = \lim_{n \rightarrow \infty} \mathbb{E}_{\mathbf{Y} \sim P_{\mathbf{Y}}} \left[ W_1(P_{\mathbf{X}|\mathbf{Y}}, P_{\mathbf{X}|\mathbf{Y}}^{\mathbf{g}_n^*}) \right] = 0.$$

This **does not imply**

$$\lim_{n \rightarrow \infty} W_1(P_{\mathbf{X}|\mathbf{Y}}, P_{\mathbf{X}|\mathbf{Y}}^{\mathbf{g}_n^*}) = 0 \quad (\iff P_{\mathbf{X}|\mathbf{Y}}^{\mathbf{g}_n^*} \xrightarrow{\text{weak}} P_{\mathbf{X}|\mathbf{Y}})$$

Can be proved only up to a subsequence! Thus, statistics (mean, variance, moments, etc) converge only subsequentially!

Key elements:

- ▶ We require  $d : \Omega_x \times \Omega_y \rightarrow \mathbb{R}$  to satisfy  $d \in \text{Lip}_{xy}$ , i.e. 1-Lipschitz in  $x$  and  $y$ .
- ▶ Then we can prove that

$$g^* = \arg \min_g W_1(P_{XY}, P_{XY}^g)$$

where  $P_{XY}^g = P_{X|Y}^g P_Y$ . Thus, we are approximating the joint density!

Key elements:

- ▶ We require  $d : \Omega_x \times \Omega_y \rightarrow \mathbb{R}$  to satisfy  $d \in \text{Lip}_{xy}$ , i.e. 1-Lipschitz in  $x$  and  $y$ .
- ▶ Then we can prove that

$$g^* = \arg \min_g W_1(P_{\mathbf{X}\mathbf{Y}}, P_{\mathbf{X}\mathbf{Y}}^g)$$

where  $P_{\mathbf{X}\mathbf{Y}}^g = P_{\mathbf{X}|\mathbf{Y}}^g P_{\mathbf{Y}}$ . Thus, we are approximating the joint density!

- ▶ If we can find the sequence  $\{(g_n^*, d_n^*)\}_n$  such that

$$\lim_{n \rightarrow \infty} \Pi(g_n^*, d_n^*) = \lim_{n \rightarrow \infty} W_1(P_{\mathbf{X}\mathbf{Y}}, P_{\mathbf{X}\mathbf{Y}}^{g_n^*}) = 0,$$

then we weakly converge to the true joint density,  $P_{\mathbf{X}\mathbf{Y}}^{g_n^*} \xrightarrow{\text{weak}} P_{\mathbf{X}\mathbf{Y}}$

$$\iff \lim_{n \rightarrow \infty} \mathbb{E}_{P_{\mathbf{X}\mathbf{Y}}^{g_n^*}} [\ell(\mathbf{X}, \mathbf{Y})] = \mathbb{E}_{P_{\mathbf{X}\mathbf{Y}}} [\ell(\mathbf{X}, \mathbf{Y})] \quad \forall \ell \in C_b(\Omega_x \times \Omega_y).$$

\* *Solution of physics-based inverse problems using conditional generative adversarial networks with full gradient penalty* (R., Esandi, Dasgupta, Oberai); CMAME, 2023.

But our goal is to approximate the conditional density, or rather estimate the conditional expectations

$$\mathbb{E}_{P_{\mathbf{X}|\mathbf{Y}}} [q(\mathbf{X})|\mathbf{y}].$$

But our goal is to approximate the conditional density, or rather estimate the conditional expectations

$$\mathbb{E}_{P_{\mathbf{X}|\mathbf{Y}}} [q(\mathbf{X})|\mathbf{y}].$$

The following result shows us how to do this **robustly**:

**Theorem: R., Esandi, Dasgupta, Oberai (2023)**

Let  $\hat{\mathbf{y}}$  be such that  $P_{\mathbf{Y}}(\hat{\mathbf{y}}) \neq 0$  and  $q \in C_b(\Omega_x)$ . Then, given  $\epsilon > 0$  (and under some mild assumptions), there exists  $\sigma > 0$  and an integer  $N$  such that

$$\left| \mathbb{E}_{P_{\mathbf{X}|\mathbf{Y}}} [q(\mathbf{X})|\hat{\mathbf{y}}] - \mathbb{E}_{P_{\mathbf{X}\mathbf{Y}}^{\sigma,n}} [q(\mathbf{X})] \right| < \epsilon \quad \forall n \geq N,$$

where  $P_{\mathbf{X}\mathbf{Y}}^{\sigma,n}(\mathbf{x}, \mathbf{y}) = P_{\mathbf{X}|\mathbf{Y}}^{g_n^*}(\mathbf{x}|\mathbf{y})P_{\mathbf{Y}\sigma}(\mathbf{y})$  and  $P_{\mathbf{Y}\sigma}(\mathbf{y}) \equiv N(\hat{\mathbf{y}}, \sigma^2 \mathbf{I})$ .

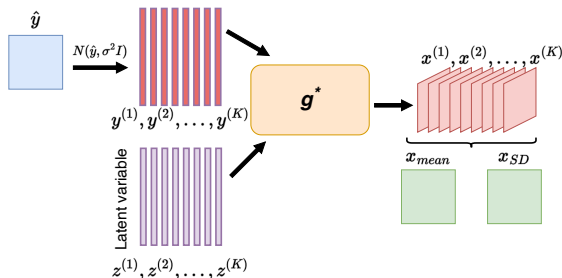
**Implication:** Instead of feeding the measurement  $\hat{\mathbf{y}}$  to  $\mathbf{g}^*$ , feed  $\hat{\mathbf{y}} + \delta_{\mathbf{y}}$  where  $\delta_{\mathbf{y}} \sim N(\mathbf{0}, \sigma^2 \mathbf{I})$

# Approximating posterior expectations

Steps:

- ▶ Acquire samples  $\mathcal{S}_x = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ , where  $\mathbf{x}_i \sim P_X^{\text{prior}}$ .
- ▶ Use forward map  $\mathcal{F}$  to generate dataset  $\mathcal{S} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_N, \mathbf{y}_N)\}$
- ▶ Train a cWGAN on  $\mathcal{S}$  but with a "full gradient penalty" term.
- ▶ For a new test measurement  $\hat{\mathbf{y}}$ , generate samples by passing  $\mathbf{z}$  and perturbed  $\mathbf{y}$  samples through  $\mathbf{g}^*$ .
- ▶ Approximate expectation using Monte Carlo.

$$\mathbb{E}_{\mathbf{X} \sim P_{\mathbf{X}|\mathbf{Y}}} [\ell(\mathbf{X})] \approx \mathbb{E}_{\hat{P}_{\mathbf{X}|\mathbf{Y}}^{g^*}} [\ell(\mathbf{X})] \approx \frac{1}{K} \sum_{i=1}^K \ell(\mathbf{g}^*(\mathbf{z}^{(i)}, \mathbf{y}^{(i)})), \quad \mathbf{z}^{(i)} \sim P_{\mathbf{Z}}, \quad \mathbf{y}^{(i)} \sim N(\hat{\mathbf{y}}, \sigma^2 \mathbf{I})$$



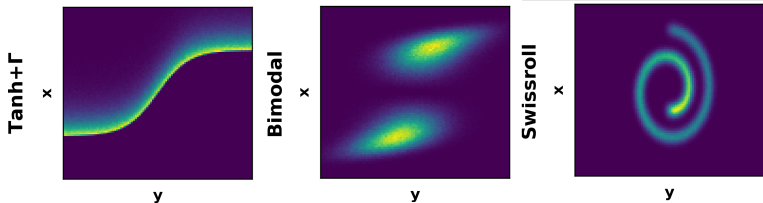
Consider the pair of 1D random variables defined by:

**Tanh +  $\Gamma$**  :  $x = \tanh(y) + \gamma$  where  $\gamma \sim \Gamma(1, 0.3)$  and  $y \sim U(-2, 2)$

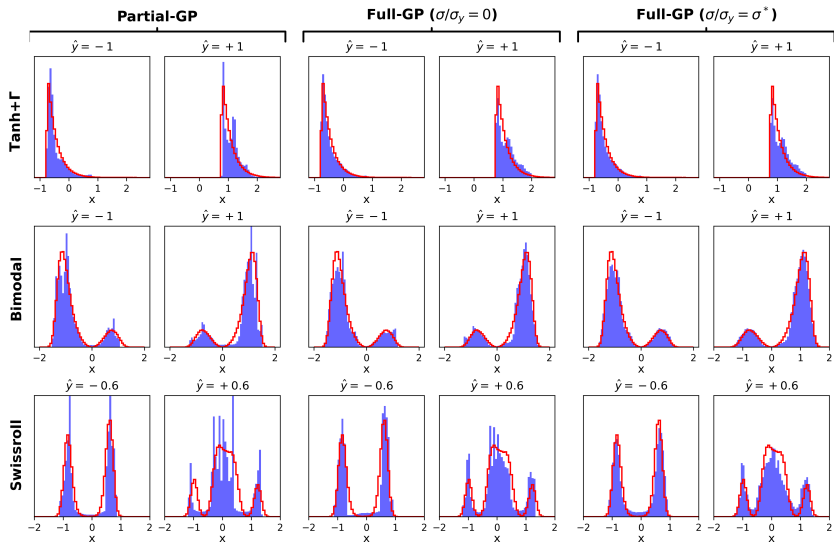
**Bimodal** :  $x = (y + w)^{1/3}$  where  $y \sim \mathcal{N}(0, 1)$  and  $w \sim \mathcal{N}(0, 1)$

**Swissroll** :  $x = 0.1t \sin(t) + 0.1w$ ,  $y = 0.1t \cos(t) + 0.1v$ ,  $t = 3\pi/2(1 + 2h)$ ,  
where  $h \sim U(0, 1)$ ,  $w \sim \mathcal{N}(0, 1)$  and  $v \sim \mathcal{N}(0, 1)$

**True joints**

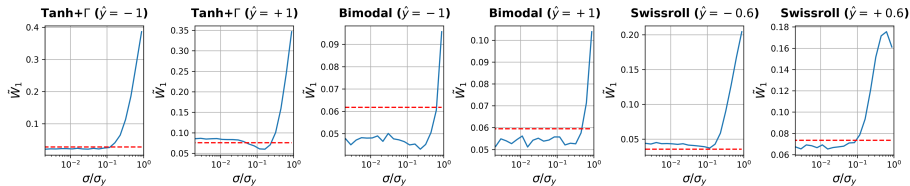


# Simple 1D problems





Errors between  $P_{X|Y}$  and  $P_{X|Y}^g$ :



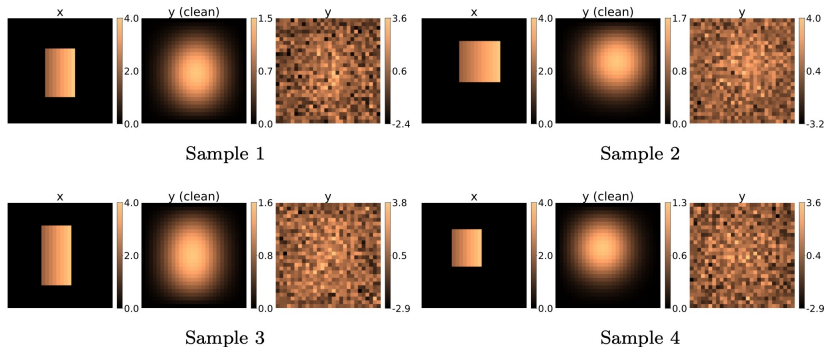
Can expect benefit of Full-GP approach on multi-modal problems!

# Solving the inverse heat conduction equation

**Goal:** Infer initial temp. field from noisy final temp. field

Assuming  $x$  to given by a rectangular inclusion and  $N_x = N_y = 28 \times 28 = 784$

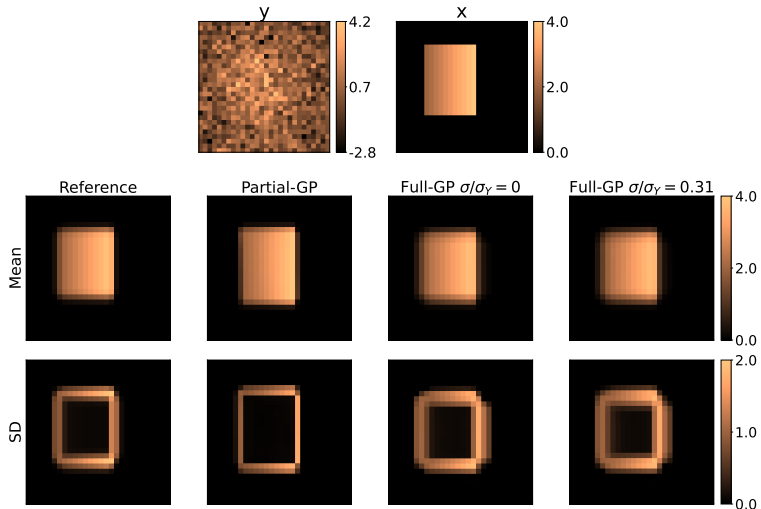
Training samples:



cWGANs trained using latent dimension  $N_z = 3!$

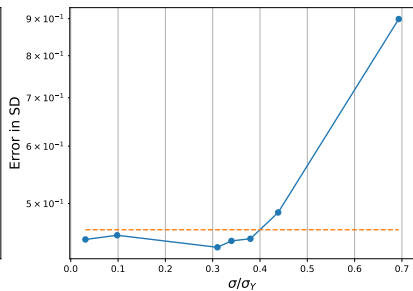
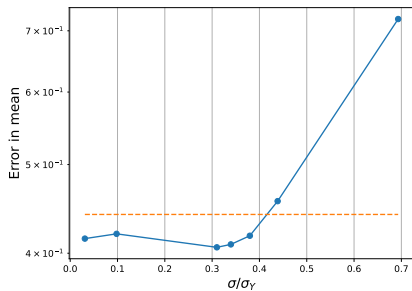
# Solving the inverse heat conduction equation

Test sample, whose reference mean as SD are available



# Solving the inverse heat conduction equation

$L^2$  error in mean and SD



- ▶ Substantial increase in wildfire activity around the globe.
- ▶ Complicated physics coupling atmosphere and wildfire dynamics.
- ▶ Correct initial state of wildfire and atmosphere variables required for successful simulations.
- ▶ Mandel et al. (2012) found that
  - Precise wildfire history during initial spread – key for model initialization.
  - History well represented by arrival time map.

**Data assimilation problem:** Given satellite measurements of active fire during initial spread, determine high resolution fire arrival map for initial period.

\* *Generative Algorithms for Fusion of Physics-Based Wildfire Spread Models with Satellite Data for Initializing Wildfire Forecasts* (Shaddy, R., Faruell, Calaza, Mandel, Haley, Hilburn, Mallia, Kochanski, Oberai); preprint on arXiv, 2023.

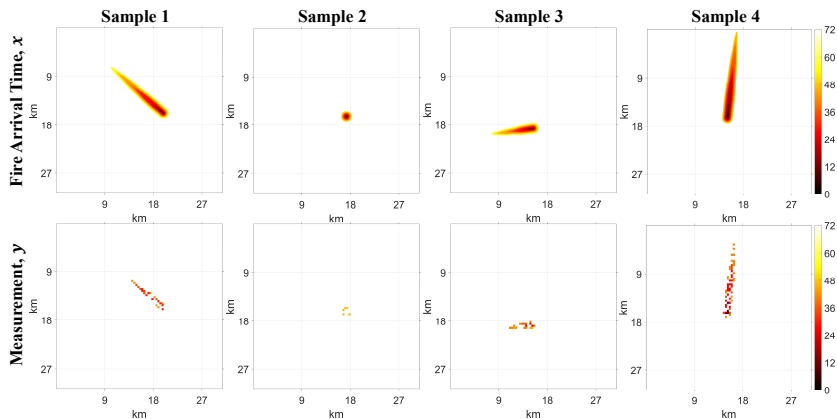
We make use of WRF-SFIRE: combines a weather forecast model with a fire-spread model.

Strategy:

- ▶ Generated 20 fire simulations using WRF-SFIRE
- ▶ Data augmented by rotations and translations to generate 10,000 high-resolution arrival maps  $\mathbf{x}_i \in \mathbb{R}^{512 \times 512}$
- ▶ Corresponding measurements  $\mathbf{y}_i \in \mathbb{R}^{512 \times 512}$  obtained by coarsening and occluding.
- ▶ 8000 training samples, 2000 validation samples (to tune hyper-parameters)

# Predicting arrival times for wildfire spread

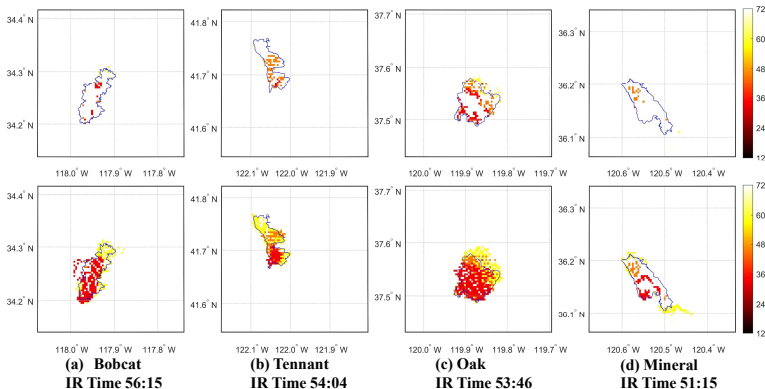
Fire arrived first in the darkest regions of the plot



Trained cWGAN with full GP and  $N_z = 100$

# Predicting arrival times for wildfire spread

- ▶ Tested on real wildfire data for fires in California between 2020 - 2022.
- ▶ Data collected from Suomi-NPP satellite, detections 2-4 times a day.
- ▶ High confidence measurements (top row); high+nominal confidence measurements (bottom row)



IR Time: Number of hours since start of fire.

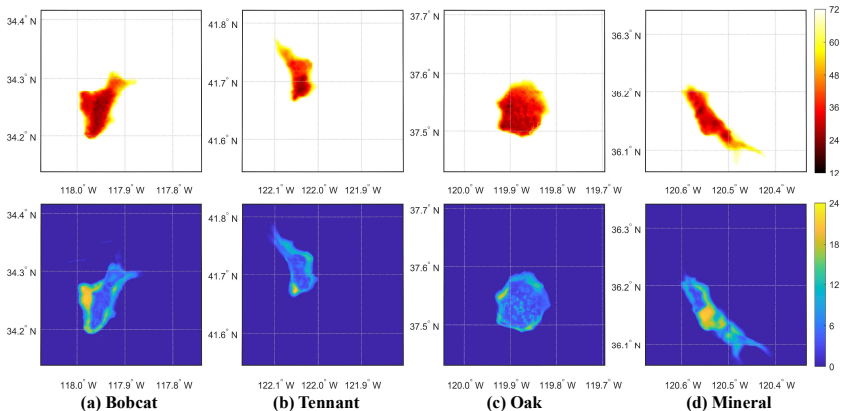


# Predicting arrival times for wildfire spread

- ▶ 200 realization for each type of input measurement.
- ▶ Weighted combination of realizations

$$\mathbf{x}_i = 0.2 \times \mathbf{x}_i^{\text{high}} + 0.8 \times \mathbf{x}_i^{\text{high+nom}}$$

used to compute pixel-wise mean and SD



Estimate ignition time based on smallest arrival time compared with California Department of Forestry and Fire Protection (CAL FIRE) reporting and another SVM based method by Farguell et al. (2021).

Wildfire	CAL FIRE	cWGAN	SVM	cWGAN Offset	SVM Offset
Tennant	23:07	<b>23 : 48</b>	21:11	<b>41 minutes</b>	1 hour 56 minutes
Oak	21:10	<b>21 : 30</b>	20:45	<b>20 minutes</b>	25 minutes
Mineral	23:40	<b>23 : 04</b>	27:53	<b>36 minutes</b>	4 hours 13 minutes

See preprint for additional details and comparisons (eg. F-score, false alarm ratio, etc)

- ▶ A cWGAN algorithm for Bayesian inference
- ▶ What do we gain?
  - Ability to represent and encode complex prior data.
  - Dimension reduction since  $N_z \ll N_x$ .
  - Sampling from cGAN is quick and easy.
  - Uncertainty quantification in terms of SD.
- ▶ Need  $(x, y)$  pairs to train – supervised algorithm.
- ▶ A theoretically sound variant using full gradient penalty of the critic
- ▶ Variants of algorithm successfully used for medical imaging tasks, such as image segmentation\* and imputation\*\*.

\* *Probabilistic Brain Extraction in MR Images via Conditional Generative Adversarial Networks* (Moazami, R., Pelletier, Oberai); IEEE Trans. Med. Imaging, 2023.

\*\* *Conditional Generative Learning for Medical Image Imputation* (Raad, R., Hwang, Gill, Duddalwar, Oberai); Scientific Reports, 2023.

- ▶ A cWGAN algorithm for Bayesian inference
- ▶ What do we gain?
  - Ability to represent and encode complex prior data.
  - Dimension reduction since  $N_z \ll N_x$ .
  - Sampling from cGAN is quick and easy.
  - Uncertainty quantification in terms of SD.
- ▶ Need  $(x, y)$  pairs to train – supervised algorithm.
- ▶ A theoretically sound variant using full gradient penalty of the critic
- ▶ Variants of algorithm successfully used for medical imaging tasks, such as image segmentation\* and imputation\*\*.

## Questions?

\* *Probabilistic Brain Extraction in MR Images via Conditional Generative Adversarial Networks* (Moazami, R., Pelletier, Oberai); IEEE Trans. Med. Imaging, 2023.

\*\* *Conditional Generative Learning for Medical Image Imputation* (Raad, R., Hwang, Gill, Duddalwar, Oberai); Scientific Reports, 2023.