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
Website: deepray.github.io

Supported by ARO grant W911NF2010050 and CARC-USC.

Brigham Young University
Oct 13, 2023

UNIVERSITY OF MARYLAND

Outline

- Challenges with Bayesian inference
- What are neural networks?
- Conditional Wasserstein generative adversarial networks (cWGANs)
- Deep posteriors with cWGANs
- Theoretical issues and a new formulation
- Numerical results
- Conclusion

Consider a forward problem

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

For example the **heat conduction** PDE:

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

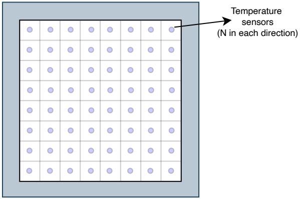
$$u(\boldsymbol{s},0) = u_0(\boldsymbol{s})$$

where

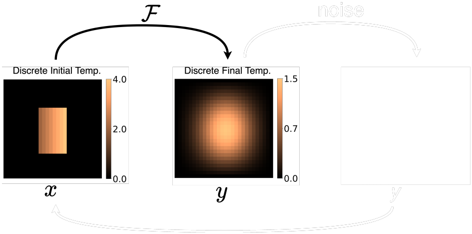
$u(\boldsymbol{s},t) \rightarrow$ temperature at location $\boldsymbol{s}$ at time t

$u_0(\boldsymbol{s}) \rightarrow$ initial temperature at location $\boldsymbol{s}$

$\kappa \rightarrow$ thermal conductivity of material



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



Forward and inverse problems

Consider a forward problem

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

For example the **heat conduction** PDE:

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

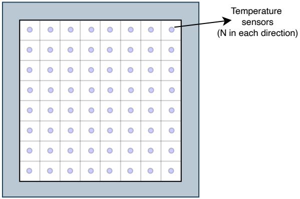
$$u(\boldsymbol{s},0) = u_0(\boldsymbol{s})$$

where

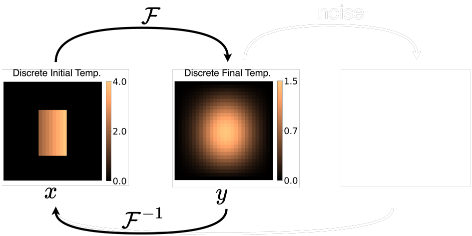
$u(\boldsymbol{s},t) \to$ temperature at location $\boldsymbol{s}$ at time t

$u_0(\boldsymbol{s}) \to$ initial temperature at location $\boldsymbol{s}$

$\kappa \to$ thermal conductivity of material



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



Consider a forward problem

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

For example the **heat conduction** PDE:

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

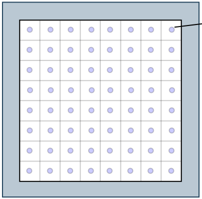
$$u(\boldsymbol{s},0) = u_0(\boldsymbol{s})$$

where

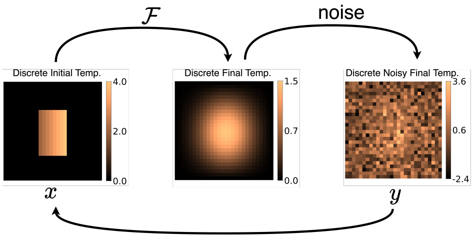
$u(\boldsymbol{s},t) \rightarrow$ temperature at location $\boldsymbol{s}$ at time t

$u_0(\boldsymbol{s}) \rightarrow$ initial temperature at location $\boldsymbol{s}$

$\kappa \rightarrow$ thermal conductivity of material



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



Challenges with inverse problems:

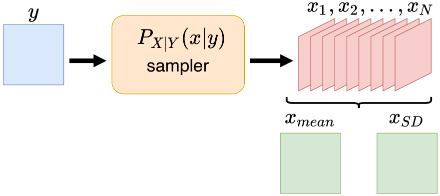
- Inverse map is not well posed.
- Noisy measurements.
- Need to encode prior knowledge about $\boldsymbol{x}$.

Bayesian framework: $\boldsymbol{x}$ and $\boldsymbol{y}$ modelled by random variables $\boldsymbol{X}$ and $\boldsymbol{Y}$.

AIM: Given a measurement $\boldsymbol{Y} = \boldsymbol{y}$ approximate the conditional (posterior) distribution

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

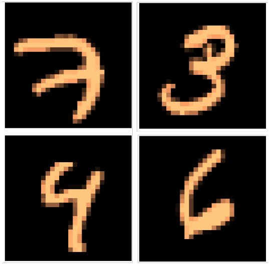
and sample from it.

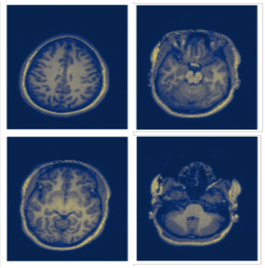


Bayesian inference: challenges

- Posterior sampling techniques, such as Markov Chain Monte Carlo, are prohibitively expensive when dimension of $\boldsymbol{X}$ 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

A neural network is a parametrized mapping

$$NN_{\boldsymbol{\psi}} : \Omega_x \to \Omega_y$$

typically formed by alternating composition

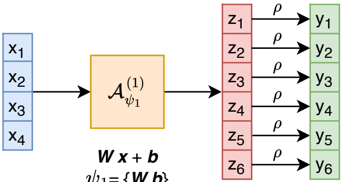
$$NN_{\boldsymbol{\psi}} := \rho \circ \mathcal{A}^{(L+1)}_{\boldsymbol{\psi}_{L+1}} \circ \rho \circ \mathcal{A}^{(L)}_{\boldsymbol{\psi}_{L}} \circ \rho \circ \mathcal{A}^{(L-1)}_{\boldsymbol{\psi}_{L-1}} \circ \cdots \circ \rho \circ \mathcal{A}^{(1)}_{\boldsymbol{\psi}_{1}}$$

where

$\boldsymbol{\psi} = \{\boldsymbol{\psi}_k\}_{k=1}^{L} \longrightarrow$ trainable weights and biases of the network

$\mathcal{A}^{(k)}_{\boldsymbol{\psi}_k} \longrightarrow$ parametrized affine transformation

$\rho \longrightarrow$ non-linear activation function



$\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$
$\psi_1 = \{\boldsymbol{W}, \boldsymbol{b}\}$

Deep neural network

A neural network is a parametrized mapping

$$NN_{\boldsymbol{\psi}} : \Omega_x \rightarrow \Omega_y$$

typically formed by alternating composition

$$NN_{\boldsymbol{\psi}} := \rho \circ \mathcal{A}^{(L+1)}_{\boldsymbol{\psi}_{L+1}} \circ \rho \circ \mathcal{A}^{(L)}_{\boldsymbol{\psi}_L} \circ \rho \circ \mathcal{A}^{(L-1)}_{\boldsymbol{\psi}_{L-1}} \circ \cdots \circ \rho \circ \mathcal{A}^{(1)}_{\boldsymbol{\psi}_1}$$

where

$$\begin{aligned} \boldsymbol{\psi} = \{\boldsymbol{\psi}_k\}_{k=1}^{L} &\longrightarrow \text{trainable weights and biases of the network} \\ \mathcal{A}^{(k)}_{\boldsymbol{\psi}_k} &\longrightarrow \text{parametrized affine transformation} \\ \rho &\longrightarrow \text{non-linear activation function} \end{aligned}$$

Usage:

- Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be related in some manner, say $\boldsymbol{y} = \mathcal{F}(\boldsymbol{x})$.
- We are only given $\mathcal{S} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$.
- $NN_{\boldsymbol{\psi}}$ can be used to learn $\mathcal{F}$.

Consider a suitable metric

$$\mu : \Omega_x \times \Omega_y \times \Omega_x \times \Omega_y \to \mathbb{R}$$

s.t. $\mu(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{x}, NN_{\boldsymbol{\psi}}(\boldsymbol{x}))$ is the error/discrepancy between $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{S}$ and $(\boldsymbol{x}, NN_{\boldsymbol{\psi}}(\boldsymbol{x}))$.

Define the loss/objective function

$$\Pi(\boldsymbol{\psi}) = \frac{1}{N} \sum_{i=1}^{N} \mu(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{x}_i, NN_{\boldsymbol{\psi}}(\boldsymbol{x}_i))$$

Solve the optimization problem

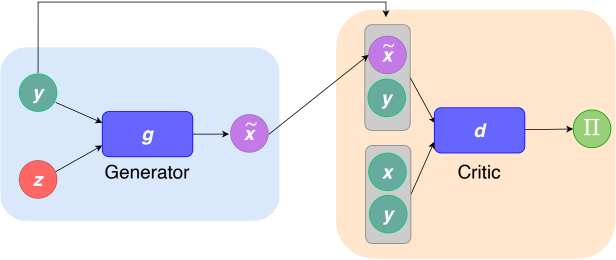
$$\boldsymbol{\psi}^* = \arg\min_{\boldsymbol{\psi}} \Pi(\boldsymbol{\psi})$$

Then $NN_{\boldsymbol{\psi}^*} \approx \mathcal{F}$

Also need to tune network hyper-parameters:

- Width • Depth (L) • Activation function ρ • Optimizer
- Loss function • Dataset

- **Notation flip** for inverse problems – infer $\boldsymbol{x}$ given $\boldsymbol{y}$!
- Proposed by Mirza et al. (2014)
- Learning distributions conditioned on another field.
- Comprises two neural networks, $\boldsymbol{g}$ and d.



Generator network:

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

Critic network:

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

Conditional Wasserstein GAN (cWGAN)

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

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

Conditional Wasserstein GAN (cWGAN)

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

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

- Objective function

$$\Pi(\boldsymbol{g}, d) = \mathop{\mathbb{E}}_{\substack{(\boldsymbol{X},\boldsymbol{Y})\sim P_{\boldsymbol{X}\boldsymbol{Y}} \\ \boldsymbol{Z}\sim P_{\boldsymbol{Z}}}} \left[d(\boldsymbol{X}, \boldsymbol{Y}) - d\big(\boldsymbol{g}(\boldsymbol{Z}, \boldsymbol{Y}), \boldsymbol{Y}\big)\right]$$

- Define

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

Conditional Wasserstein GAN (cWGAN)

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

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

- Objective function

$$\Pi(\boldsymbol{g}, d) = \mathop{\mathbb{E}}_{\substack{(\boldsymbol{X},\boldsymbol{Y})\sim P_{\boldsymbol{X}\boldsymbol{Y}} \\ \boldsymbol{Z}\sim P_{\boldsymbol{Z}}}} \left[d(\boldsymbol{X}, \boldsymbol{Y}) - d\big(\boldsymbol{g}(\boldsymbol{Z}, \boldsymbol{Y}), \boldsymbol{Y}\big) \right]$$

- Define

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

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

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

$$\boldsymbol{g}^* = \arg\min_{\boldsymbol{g}} \Pi(\boldsymbol{g}, d^*(\boldsymbol{g})).$$

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

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

  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

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

  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

$$d^*(\boldsymbol{g}) = \arg\max_{d} \left[\Pi(\boldsymbol{g}, d) - \lambda \mathcal{GP}_x \right]$$

  where

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

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

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

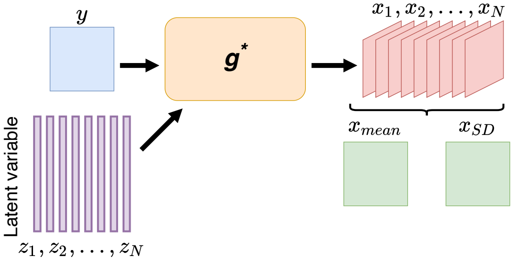
Steps:

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

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

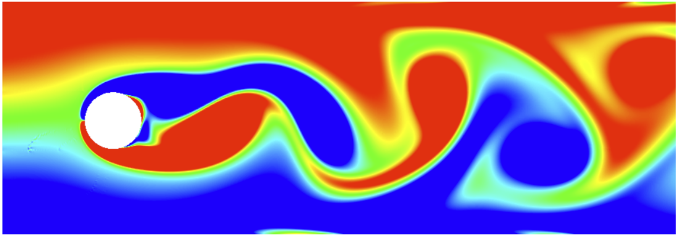
- Train a cWGAN on $\mathcal{S}$.
- For a new test measurement $\boldsymbol{y}$, generate samples using $\boldsymbol{g}^*$.
- Evaluate statistics using Monte Carlo.

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



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

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



Original model by Adler et al.

- Designed for CT imaging applications
- In $\boldsymbol{g}$, latent variable $\boldsymbol{z}$ stacked as additional channel of $\boldsymbol{y}$

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

$\longrightarrow$ N_z scales with N_y! Won't see dim. reduction

- Specialized architecture required for critic to avoid mode collapse

$\longrightarrow$ larger network and higher training cost!

Modified version*

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

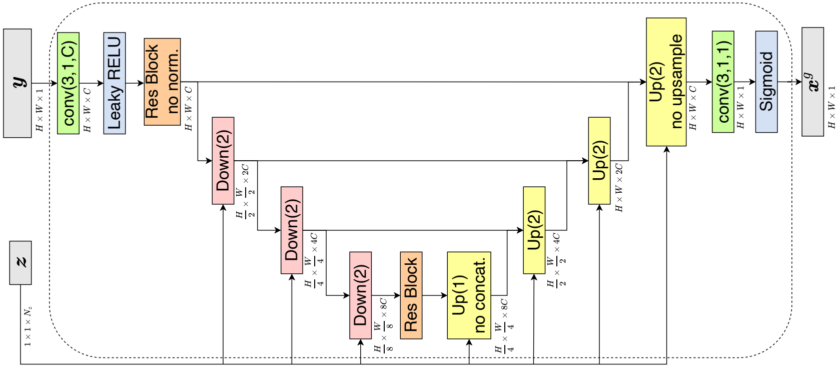
$$\boldsymbol{y} \in \mathbb{R}^{N \times M \times C}, \quad \boldsymbol{z} \in \mathbb{R}^{N_z}$$

$\longrightarrow$ N_z no longer depends on N_y – get dim. reduction!
$\longrightarrow$ 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 $\boldsymbol{x}$, $\boldsymbol{y}$ have tensored (image-like) structure.



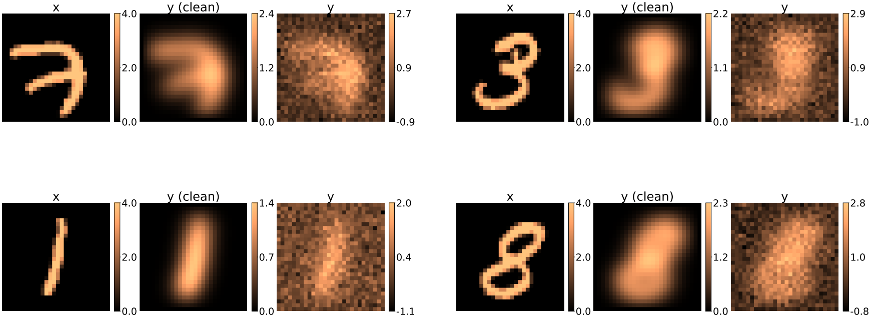
Consider the PDE

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

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

Assuming $\boldsymbol{x}$ to given by (heat stamped) MNIST handwritten digits and $N_x = N_y = 28 \times 28 = 784$

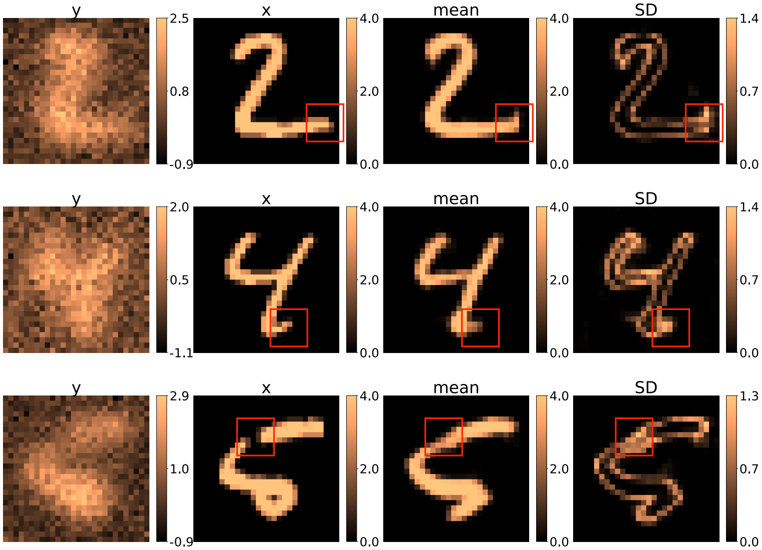
Training samples:



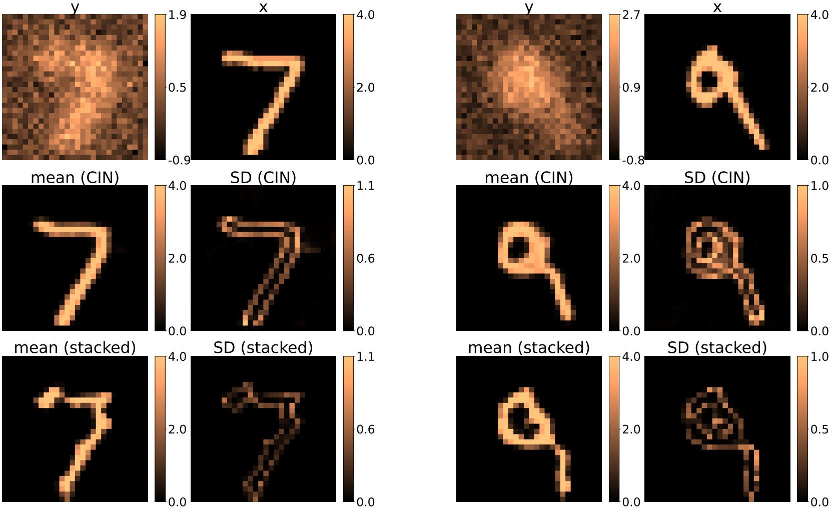
cWGAN trained using latent dimension $N_z = 100$.

SD can be used as a metric for UQ

Testing trained cGAN (statistics with $K = 800$ $\boldsymbol{z}$ samples)



Low ensemble variability and poor reconstruction with original stacked approach!



See paper for more experiments and a discussion on generalizability!

A theoretical issue with interpreting convergence

Recall that

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

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

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

A theoretical issue with interpreting convergence

Recall that

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

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

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

Consider the sequence $\{\boldsymbol{g}^*_n\}_n$, where $n \in \mathbb{Z}_+$ is the number of weights/biases, s.t.

$$\lim_{n\to\infty} \Pi(\boldsymbol{g}^*_n, d^*_n) = \lim_{n\to\infty} \mathop{\mathbb{E}}_{\boldsymbol{Y}\sim P_{\boldsymbol{Y}}} \left[W_1(P_{\boldsymbol{X}|\boldsymbol{Y}}, P^{\boldsymbol{g}^*_n}_{\boldsymbol{X}|\boldsymbol{Y}}) \right] = 0.$$

A theoretical issue with interpreting convergence

Recall that

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

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

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

Consider the sequence $\{\boldsymbol{g}^*_n\}_n$, where $n \in \mathbb{Z}_+$ is the number of weights/biases, s.t.

$$\lim_{n\to\infty} \Pi(\boldsymbol{g}^*_n, d^*_n) = \lim_{n\to\infty} \mathop{\mathbb{E}}_{\boldsymbol{Y}\sim P_{\boldsymbol{Y}}} \left[W_1(P_{\boldsymbol{X}|\boldsymbol{Y}}, P^{\boldsymbol{g}^*_n}_{\boldsymbol{X}|\boldsymbol{Y}}) \right] = 0.$$

This does not imply

$$\lim_{n\to\infty} W_1(P_{\boldsymbol{X}|\boldsymbol{Y}}, P^{\boldsymbol{g}^*_n}_{\boldsymbol{X}|\boldsymbol{Y}}) = 0 \quad \left(\iff P^{\boldsymbol{g}^*_n}_{\boldsymbol{X}|\boldsymbol{Y}} \overset{weak}{\longrightarrow} P_{\boldsymbol{X}|\boldsymbol{Y}} \right)$$

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

A novel cWGAN*

Key elements:

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

$$\boldsymbol{g}^* = \arg\min_{\boldsymbol{g}} W_1(P_{\boldsymbol{XY}}, P^{\boldsymbol{g}}_{\boldsymbol{XY}})$$

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

Key elements:

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

$$\boldsymbol{g}^* = \arg\min_{\boldsymbol{g}} W_1(P_{\boldsymbol{XY}}, P^{\boldsymbol{g}}_{\boldsymbol{XY}})$$

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

- If we can find the sequence $\{\boldsymbol{g}^*_n\}_n$ such that

$$\lim_{n\to\infty} \Pi(\boldsymbol{g}^*_n, d^*_n) = \lim_{n\to\infty} W_1(P_{\boldsymbol{XY}}, P^{\boldsymbol{g}^*_n}_{\boldsymbol{XY}}) = 0,$$

then we weakly converge to the true joint density, $P^{\boldsymbol{g}^*_n}_{\boldsymbol{XY}} \xrightarrow{weak} P_{\boldsymbol{XY}}$

$$\iff \lim_{n\to\infty} \mathop{\mathbb{E}}_{P^{\boldsymbol{g}^*_n}_{\boldsymbol{XY}}} [\ell(\boldsymbol{X}, \boldsymbol{Y})] = \mathop{\mathbb{E}}_{P_{\boldsymbol{XY}}} [\ell(\boldsymbol{X}, \boldsymbol{Y})] \quad \forall\, \ell \in C_b(\Omega_x \times \Omega_y).$$

* *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

$$\mathop{\mathbb{E}}_{P_{\boldsymbol{X}|\boldsymbol{Y}}} \left[q(\boldsymbol{X}) | \boldsymbol{y} \right].$$

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

$$\underset{P_{\boldsymbol{X}|\boldsymbol{Y}}}{\mathbb{E}}\left[q(\boldsymbol{X})|\boldsymbol{y}\right].$$

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

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

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

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

where $\hat{P}^{\boldsymbol{g}_n^*}_{\boldsymbol{X}\boldsymbol{Y}_\sigma}(\boldsymbol{x},\boldsymbol{y}) = P^{\boldsymbol{g}_n^*}_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y})P_{\boldsymbol{Y}_\sigma}(\boldsymbol{y})$ and $P_{\boldsymbol{Y}_\sigma}(\boldsymbol{y}) \equiv N(\hat{\boldsymbol{y}}, \sigma^2\boldsymbol{I})$.

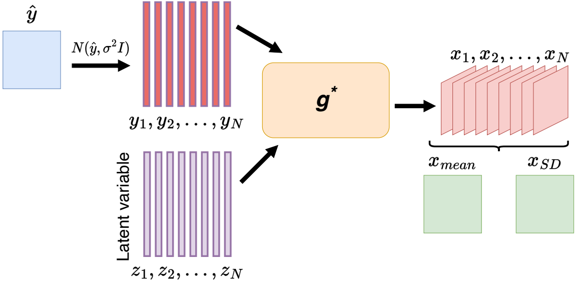
Proof uses the fact that the Dirac measure can be approximated by Gaussians.

Approximating posterior expectations

Steps:

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

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

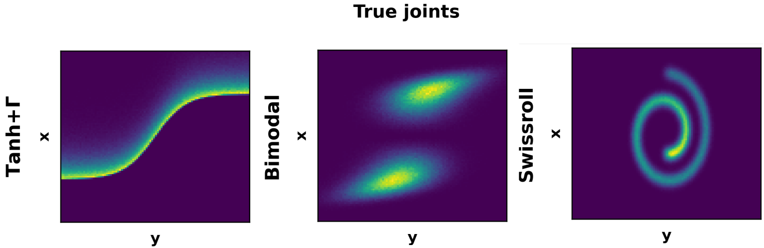


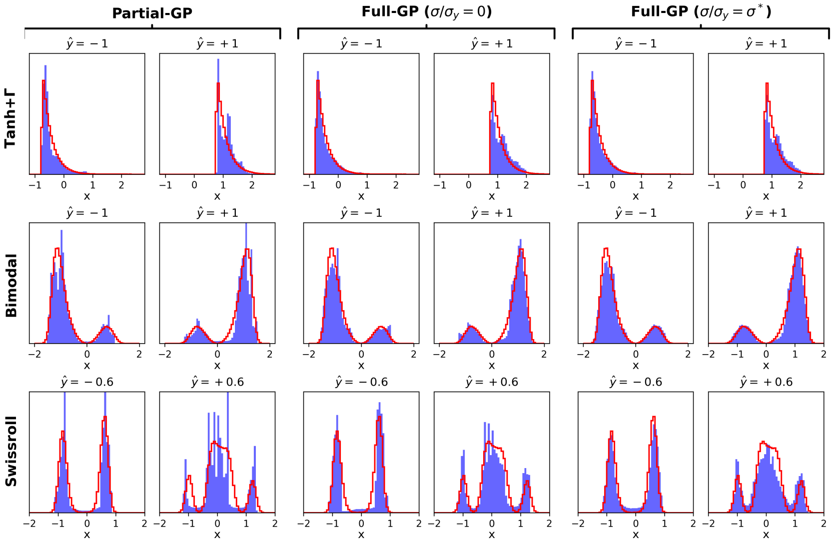
Consider the pair of 1D random variables defined by:

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

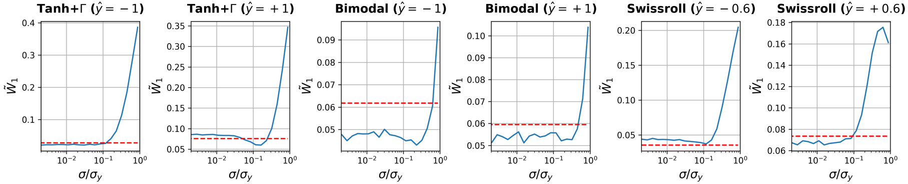
Bimodal : $x = (y + w)^{1/3}$ where $y \sim \mathcal{N}(0, 1)$ and $w \sim \mathcal{N}(0, 1)$

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 $v \sim \mathcal{N}(0, 1)$




Partial-GP
Full-GP ($\sigma/\sigma_y = 0$)
Full-GP ($\sigma/\sigma_y = \sigma^*$)
Tanh+Γ
Bimodal
Swissroll
$\hat{y} = -1$
$\hat{y} = +1$
$\hat{y} = -0.6$
$\hat{y} = +0.6$
x

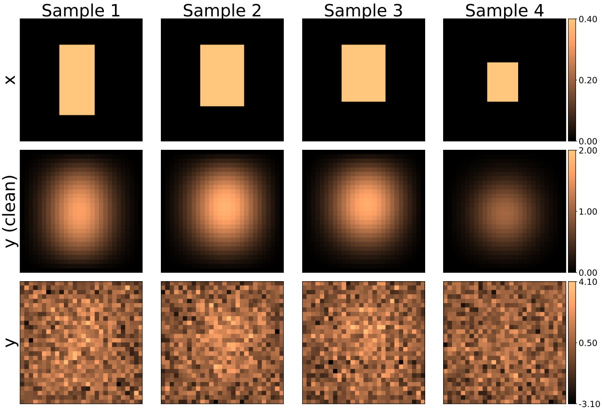
Errors between $P_{X|Y}$ and $P^g_{X|Y}$:



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

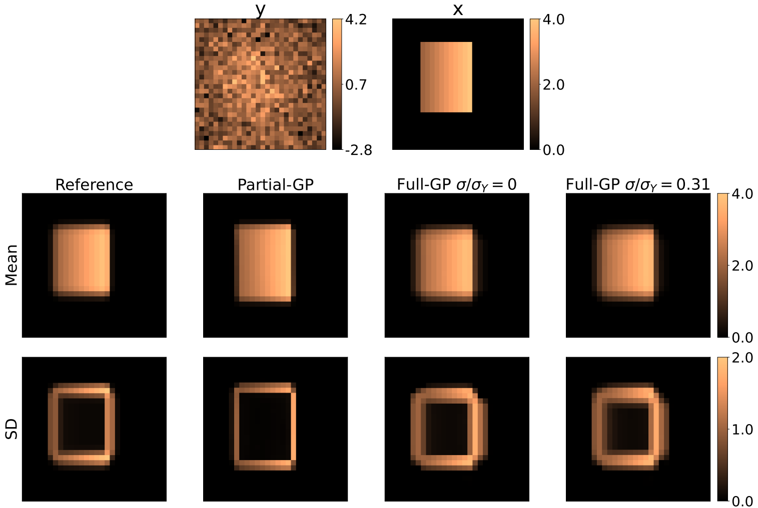
Goal: Infer initial temp. field from noisy final temp. field
Assuming $\boldsymbol{x}$ to given by a rectangular inclusion and $N_x = N_y = 28 \times 28 = 784$

Training samples:

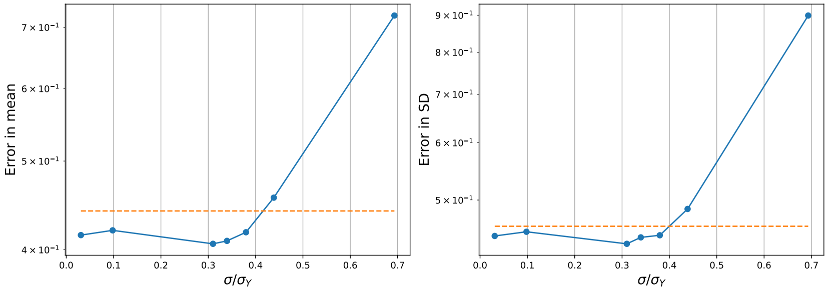


cWGANs trained using latent dimension $N_z = 3$!

Test sample, whose reference mean as SD are available



L^2 error in mean and SD



Predicting arrival times for wildfire spread*

- 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.

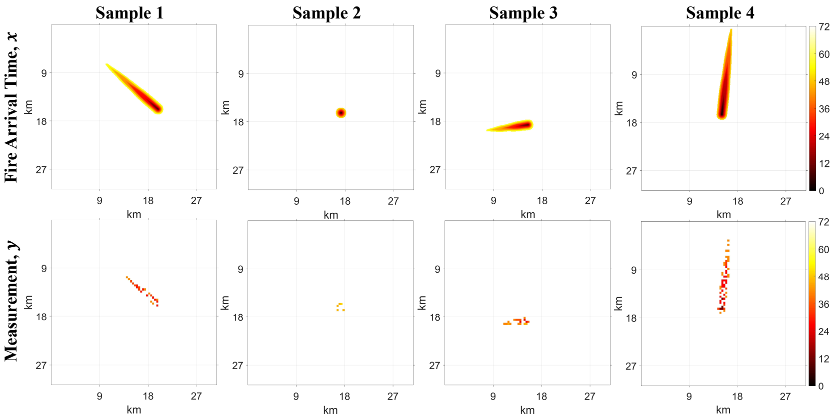
* *Generative Algorithms for Fusion of Physics-Based Wildfire Spread Models with Satellite Data for Initializing Wildfire Forecasts* (Shaddy, R., Faruell, Calaza, Mandel, Haley, Hilburn, Mallia, Kochanski, Oberai); preprint on arXiv, 2023.

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

Strategy:

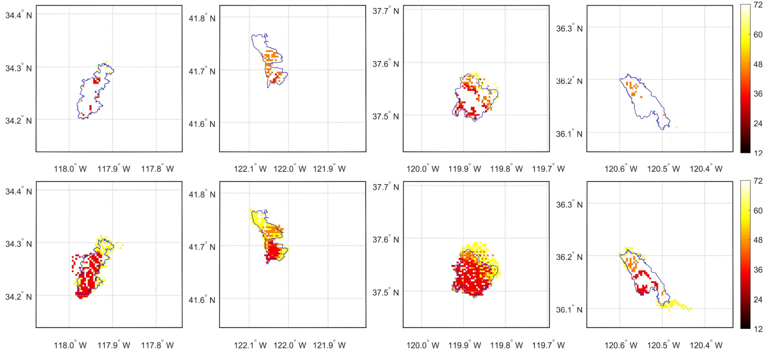
- Generated 20 fire simulations using WRF-SFIRE
- Data augmented by rotations and translations to generate 10,000 high-resolution arrival maps $\boldsymbol{x}_i \in \mathbb{R}^{512\times512}$
- Corresponding measurements $\boldsymbol{y}_i \in \mathbb{R}^{512\times512}$ obtained by coarsening and occluding.
- 8000 training samples, 2000 validation samples (to tune hyper-parameters)

Fire arrived first in the darkest regions of the plot



Trained cWGAN with full GP and $N_z = 100$

- 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)



(a) Bobcat **IR Time 56:15**

(b) Tennant **IR Time 54:04**

(c) Oak **IR Time 53:46**

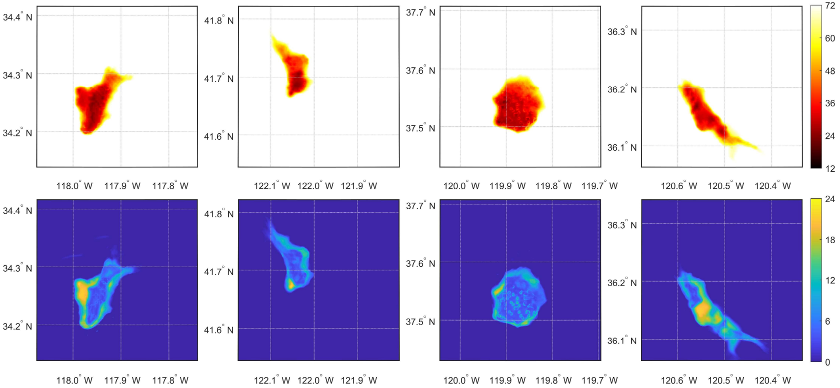
(d) Mineral **IR Time 51:15**

IR Time: Number of hours since start of fire.

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

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

used to compute pixel-wise mean and SD



(a) Bobcat **(b) Tennant** **(c) Oak** **(d) Mineral**

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 Error	SVM Error
Tennant	23:07	$\mathbf{23:48}$	21:11	$\mathbf{41\ minutes}$	1 hour 56 minutes
Oak	21:10	$\mathbf{21:30}$	20:45	$\mathbf{20\ minutes}$	25 minutes
Mineral	23:40	$\mathbf{23:04}$	27:53	$\mathbf{36\ minutes}$	4 hours 13 minutes

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

Conclusion

- 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 $(\boldsymbol{x}, \boldsymbol{y})$ pairs to train – supervised algorithm.
- A theoretically sound variant using full gradient penalty of the ciritic
- Currently testing algorithm on several other physics-based and medical applications.

Questions?