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.

Data Science Symposium
SDSU February 6, 2024

Motivation: 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:

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

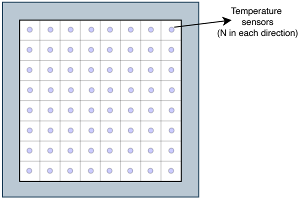
$$u(\boldsymbol{s}, 0) = u_0(\boldsymbol{s})$$

where

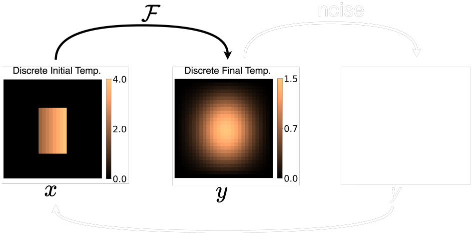
$u(\boldsymbol{s}, t) \rightarrow$ temperature at location $\boldsymbol{s}$ at time t

$u_0(\boldsymbol{s}) \rightarrow$ 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 $u(\boldsymbol{s}, T)$



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:

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

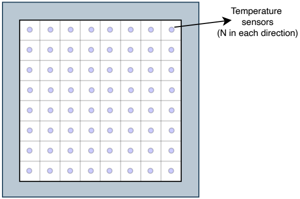
$$u(\boldsymbol{s}, 0) = u_0(\boldsymbol{s})$$

where

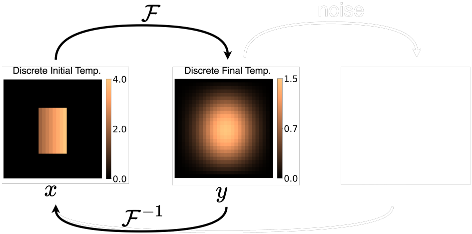
$u(\boldsymbol{s}, t) \rightarrow$ temperature at location $\boldsymbol{s}$ at time t

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

$\kappa \rightarrow$ thermal conductivity of material



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

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

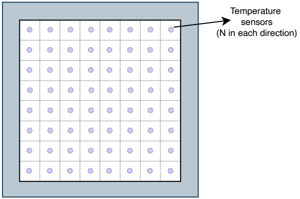
$$u(\boldsymbol{s},0) = u_0(\boldsymbol{s})$$

where

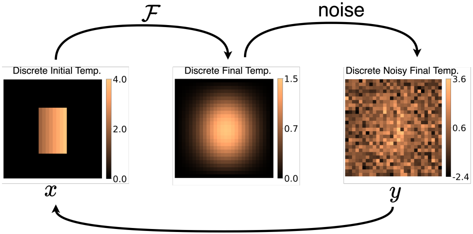
$u(\boldsymbol{s},t) \rightarrow$ temperature at location $\boldsymbol{s}$ at time t

$u_0(\boldsymbol{s}) \rightarrow$ 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:

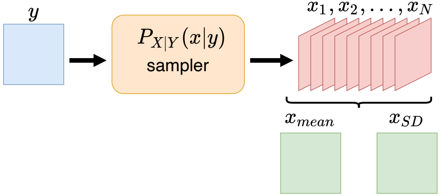
- Inverse map is not well posed.
- Noisy measurements.
- Need to encode prior knowledge about $\boldsymbol{x}$.

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

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

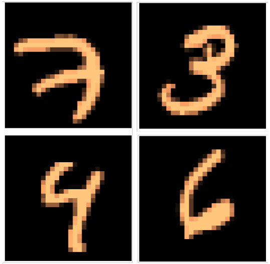
and sample from it.

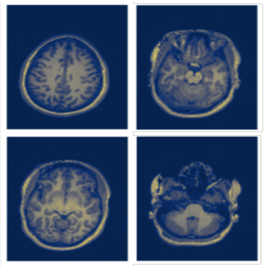


Bayesian inference: challenges

- 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, $\boldsymbol{x}$ data might look like:

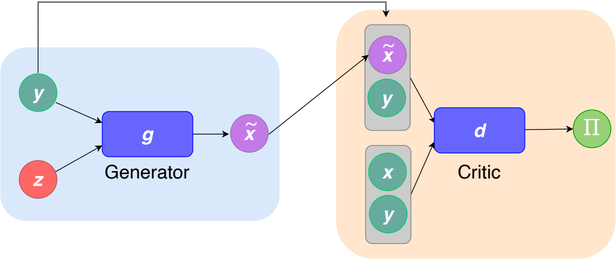




Representing this data using simple distributions is hard!

Resolve both issues using deep learning

- Proposed by Mirza et al. (2014)
- Learning distributions conditioned on another field ($P_{\boldsymbol{X}|\boldsymbol{Y}}$)
- Comprises two neural networks, $\boldsymbol{g}$ and d.



Generator network:

- $\boldsymbol{g} : \Omega_z \times \Omega_y \to \Omega_x$.
- 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 $P_{\boldsymbol{X}\boldsymbol{Y}}$

Critic network:

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

- 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

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

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

A Novel Wasserstein cGAN *

- Objective function

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

- Define

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

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

$$d^*(\boldsymbol{g}) = \arg\max_{d \in \text{Lip}} \Pi(\boldsymbol{g}, d) \quad (= W_1(P_{\boldsymbol{XY}}, P^{\boldsymbol{g}}_{\boldsymbol{XY}}))$$
$$\boldsymbol{g}^* = \arg\min_{\boldsymbol{g}} \Pi(\boldsymbol{g}, d^*(\boldsymbol{g})).$$

where $P^{\boldsymbol{g}}_{\boldsymbol{XY}} = P^{\boldsymbol{g}}_{\boldsymbol{X}|\boldsymbol{Y}} P_{\boldsymbol{Y}}$.

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

- Assume there exists a sequence $\{(\boldsymbol{g}_n^*, d_n^*)\}_n$ such that

$$\lim_{n\to\infty} \Pi(\boldsymbol{g}_n^*, d_n^*) = 0,$$

- Let $\hat{\boldsymbol{y}}$ be such that $P_{\boldsymbol{Y}}(\hat{\boldsymbol{y}}) \neq 0$ and $q \in C_b(\Omega_x)$.

- Assume there exists a sequence $\{(\boldsymbol{g}_n^*, d_n^*)\}_n$ such that

$$\lim_{n\to\infty} \Pi(\boldsymbol{g}_n^*, d_n^*) = 0,$$

- Let $\hat{\boldsymbol{y}}$ be such that $P_{\boldsymbol{Y}}(\hat{\boldsymbol{y}}) \neq 0$ and $q \in C_b(\Omega_x)$.

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

Given $\epsilon > 0$ (and under some mild assumptions), there exists a $\sigma > 0$ and an integer N such that

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

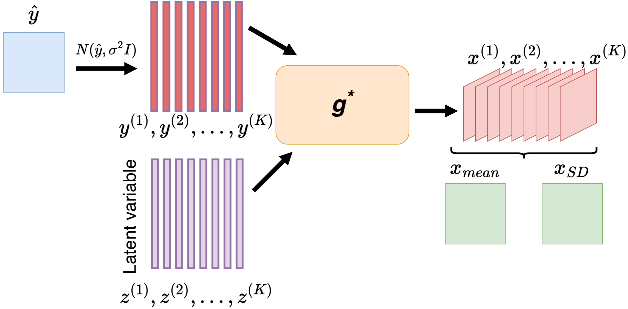
where $P^{\sigma,n}_{\boldsymbol{X}\boldsymbol{Y}}(\boldsymbol{x}, \boldsymbol{y}) = P^{\boldsymbol{g}_n^*}_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y}) P_{\boldsymbol{Y}_\sigma}(\boldsymbol{y})$ and $P_{\boldsymbol{Y}_\sigma}(\boldsymbol{y}) \equiv N(\hat{\boldsymbol{y}}, \sigma^2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

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

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 $\mathcal{S}$.
- For a test $\hat{\boldsymbol{y}}$, generate samples by passing $\boldsymbol{z}$ and “perturbed" $\boldsymbol{y}$ samples through $\boldsymbol{g}^*$.
- Approximate expectation using Monte Carlo.

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

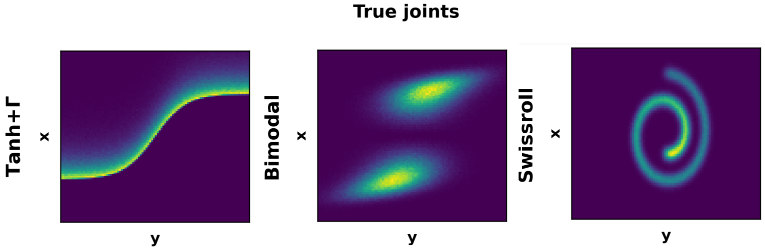


Consider the pair of 1D random variables defined by:

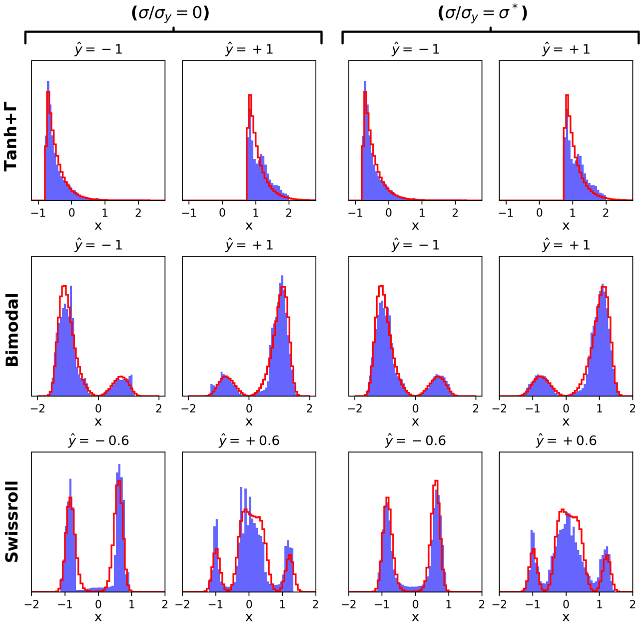
Tanh + Γ : $x = \tanh(y) + \gamma$ where $\gamma \sim \Gamma(1, 0.3)$ and $y \sim U(-2, 2)$

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



Simple 1D problems



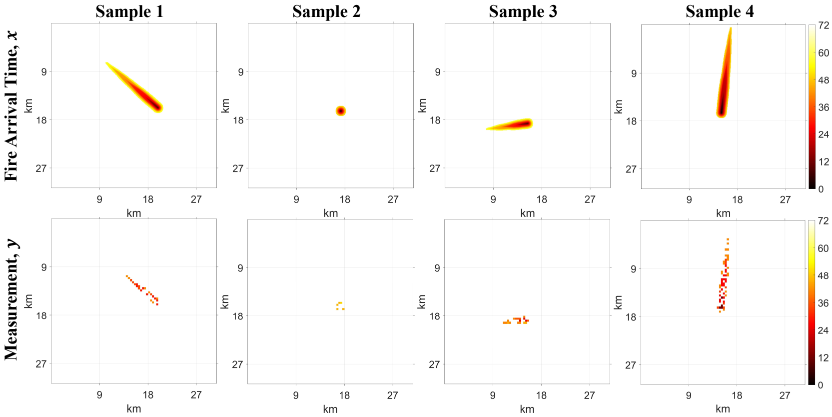
Predicting arrival times for wildfire spread*

- 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.

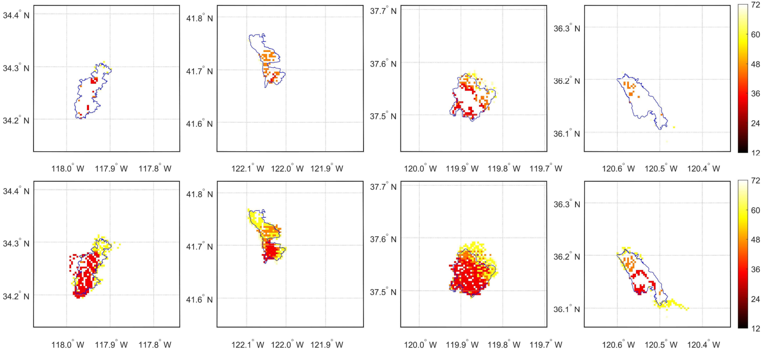
* *Generative Algorithms for Fusion of Physics-Based Wildfire Spread Models with Satellite Data for Initializing Wildfire Forecasts* (Shaddy, R., Faruell, Calaza, Mandel, Haley, Hilburn, Mallia, Kochanski, Oberai); preprint on arXiv, 2023.

- Synthetic training data generated using **WRF-SFIRE**.
- $N_x = N_y = 512 \times 512 = 262,144$.
- Trained cWGAN $N_z = 100$.



Fire arrived first in the darkest regions of the plot

- 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)



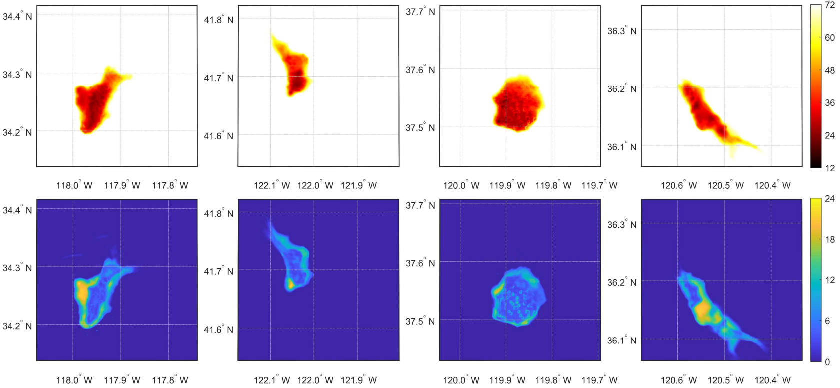
(a) Bobcat IR Time 56:15 **(b) Tennant IR Time 54:04** **(c) Oak IR Time 53:46** **(d) Mineral IR Time 51:15**

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)

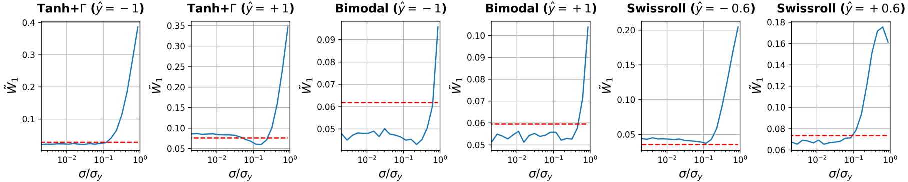
Conclusion

- 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.
- Convergence theory that leads to a robust sampling algorithm.
- Currently testing algorithm on several other physics-based and medical applications.

Questions?

Simple 1D problems: role of σ

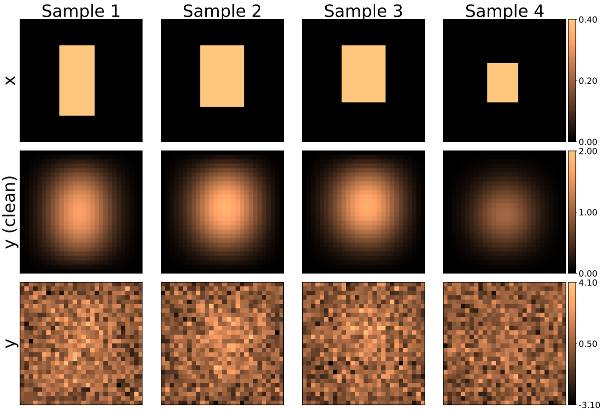
Errors between $P_{X|Y}$ and $P^g_{X|Y}$:



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

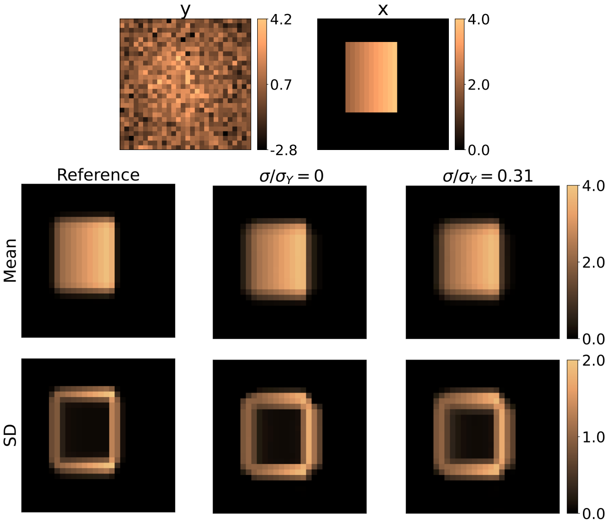
Goal: Infer initial temp. field from noisy final temp. field
Assuming $\boldsymbol{x}$ to given by a rectangular inclusion and $N_x = N_y = 28 \times 28 = 784$

Training samples:

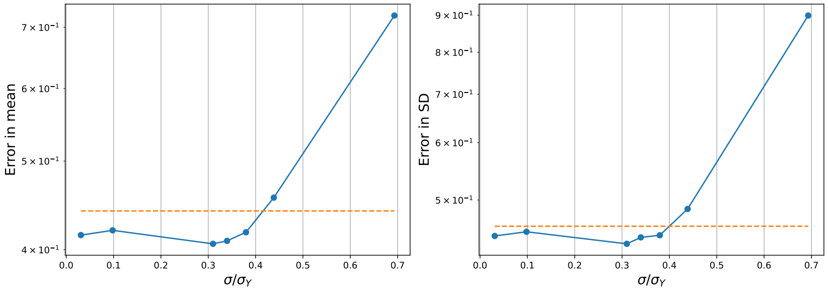


cWGANs trained using latent dimension $N_z = 3$!

Test sample, whose reference mean as SD are available



L^2 error in mean and SD



WRF-SFIRE simulation initialization

- 20 simulations in total
- 2 day fire spread over flat terrain with a uniform fuel type of brush.
- Initial wind profile: logarithmic profile up to 2 km, constant wind speed above 2 km, uniform prescription in one direction
- Wind magnitude 10m from the surface varied randomly from a uniform distribution between $0 - 5$ m s^{-1}.

Strategy:

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