Reconstructing PDFs from LQCD data: GP and INN

HUGS 2024

Yamil Cahuana Medrano

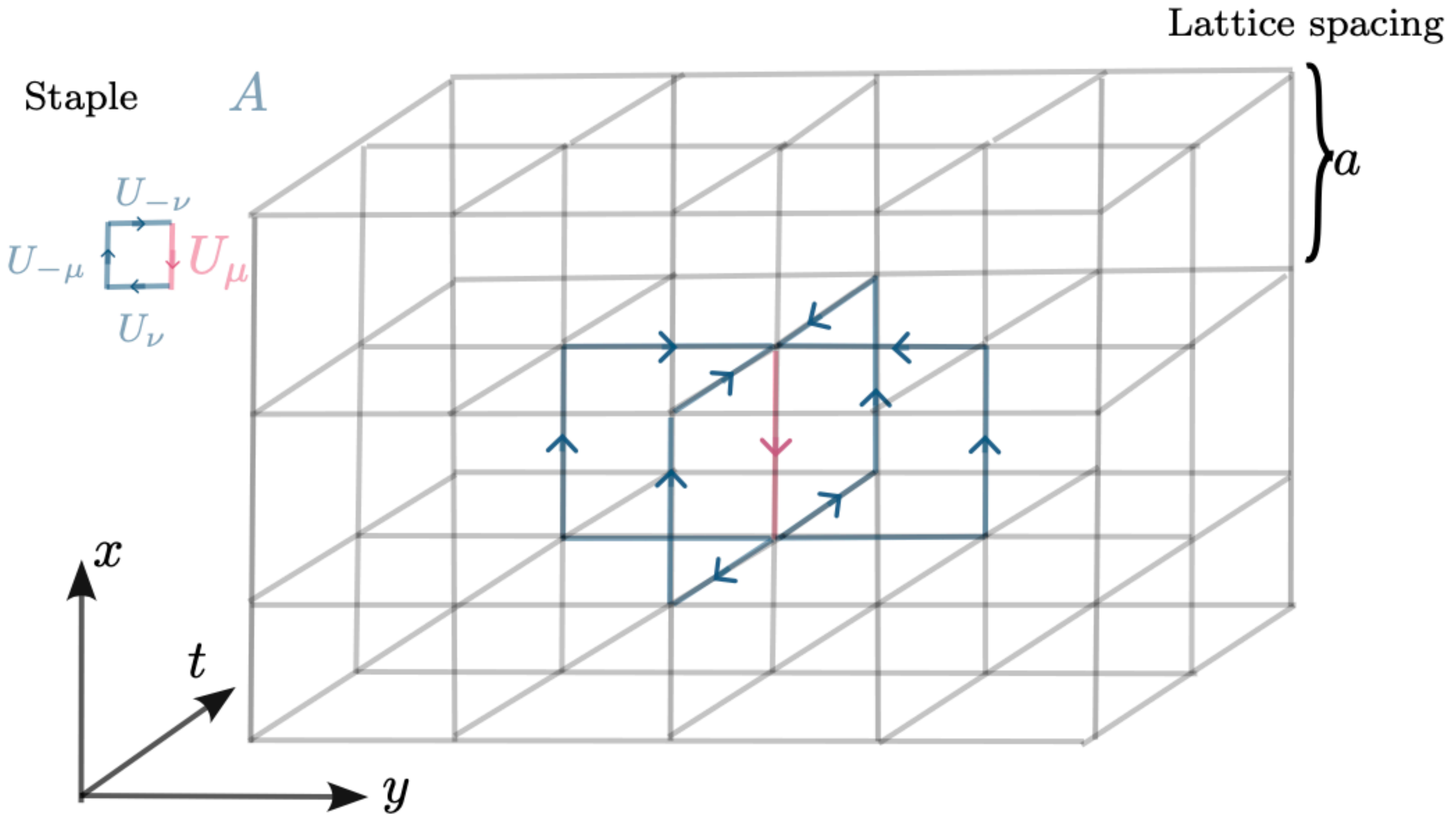
Jefferson Lab

Motivation

PDF ↔ LQCD

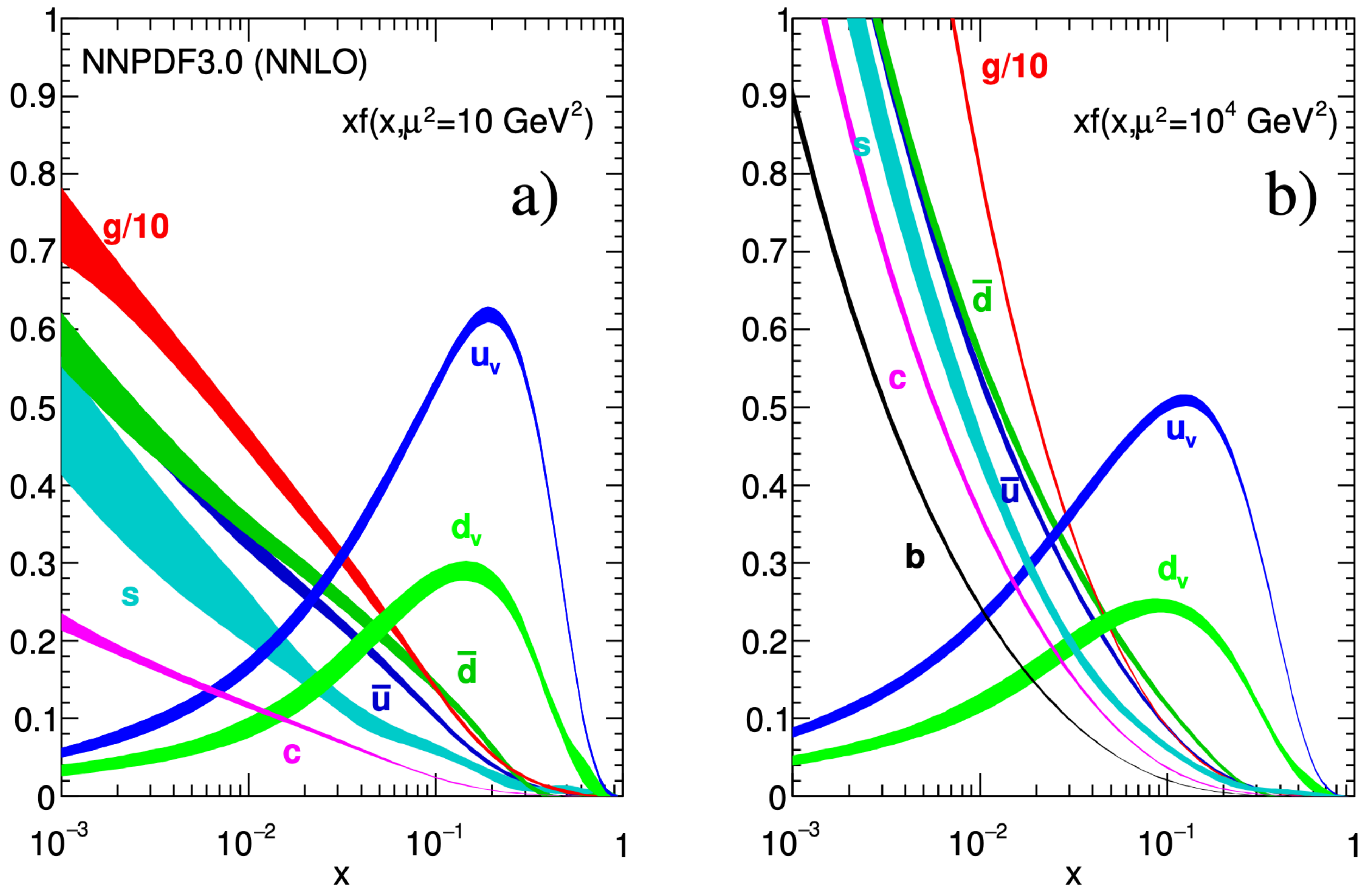
My Introduction to LQCD…
and PDFs

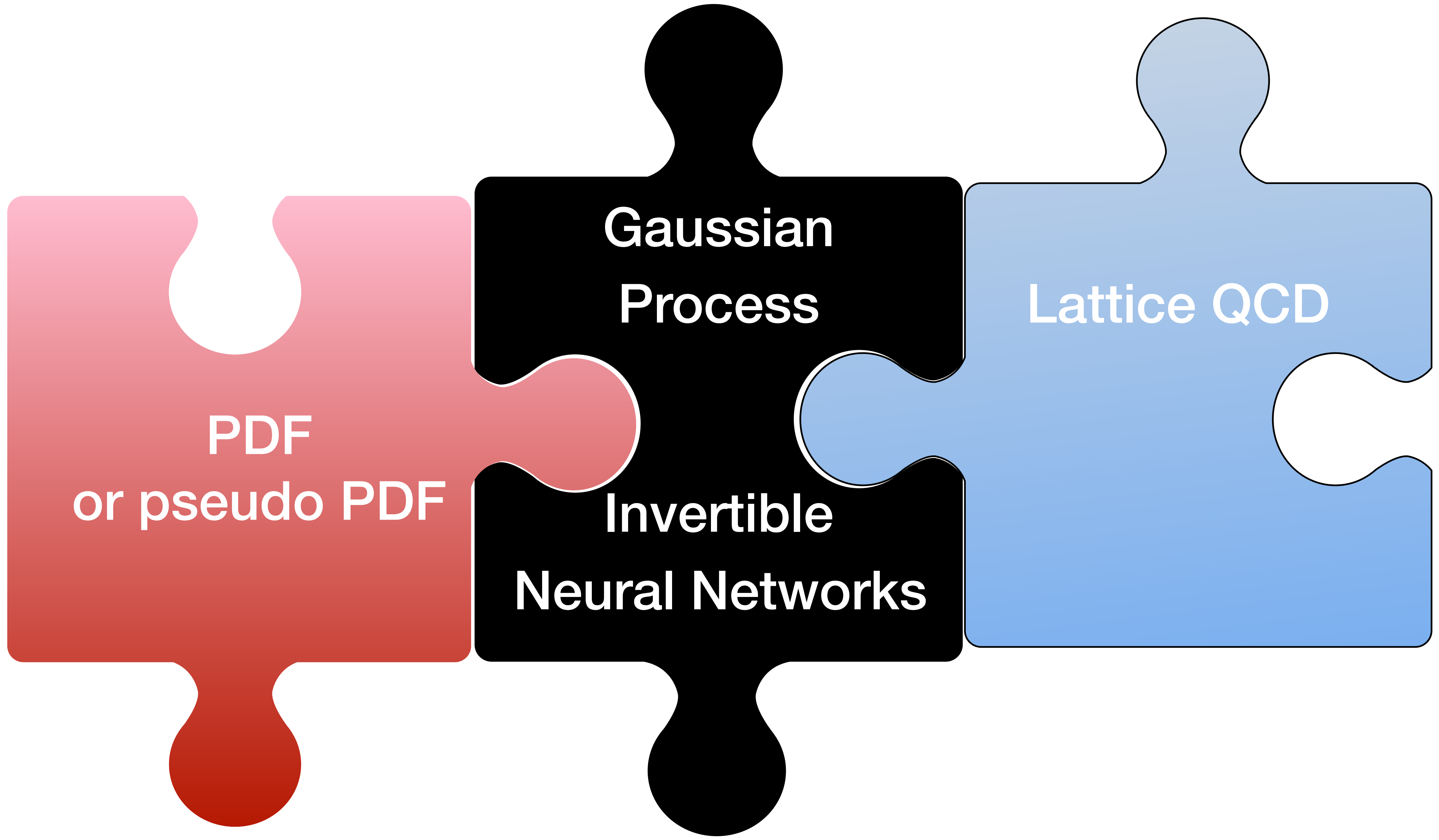
$$\mathscr{L}_{QCD} = -\frac{1}{4}F^{\alpha}_{\mu\nu}F^{\mu\nu}_{\alpha} + \bar{\psi}_i(i\not{D} - m_i)\psi_i$$



`Parton distributions and lattice QCD calculations
arXiv:1711.07916v3

Parton distributions for the LHC run II
10.1007/jhep04(2015)040





PDFs on Euclidean Lattice

Pseudo-PDFs, Ioffe-time

$\nu = p \cdot z$

X. Ji, Phys. Rev. Lett. 110, 262002 (2013).
A.Radyushkin, Phys. Rev. D 96, 034025 (2017)

- Lattice QCD prevents calculations of matrix elements on the light cone.

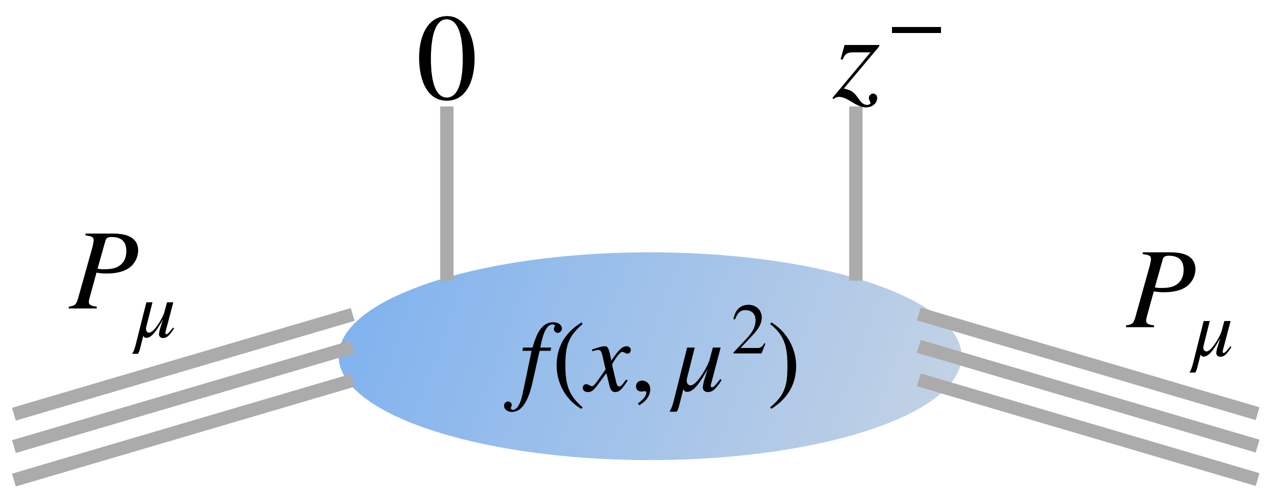
Pseudo-ITD

Lorentz decomposition $M^{\alpha}(p,z) = < p|\bar{\psi}(z)\gamma^{\alpha}U(z;0)\psi(0)|p > = p^{\alpha}\mathcal{M}(\nu,z^2) + z^{\alpha}\mathcal{N}(\nu,z^2)$

Fix the vectors in the light cone coordinates to get the pseudo ITD

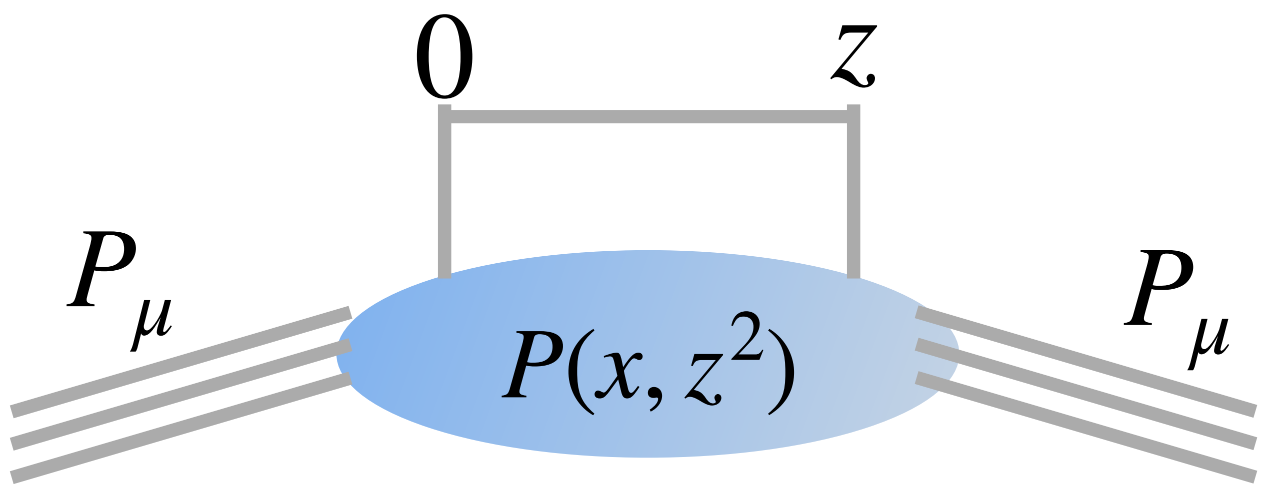
$\alpha = + \qquad z_{\alpha} = (0,z^{-},0_T) \qquad p_{\alpha} = (p^{+},\frac{m^2}{2p^{+}},0_T)$

$$\mathcal{M}(-p_{+}z_{-},0) = \int_{-1}^{1} dx f(x) e^{-ixp_{+}z_{-}}$$



- On lattice, the reduced pseudo-ITD can be extracted (Fourier-transform of the pseudo-PDF) and extrapolated to $z^2 \to 0$

$\alpha = 0 \qquad z_{\alpha} = (0,0,0,z_3) \qquad p_{\alpha} = (p_0,0,0,p_3) \qquad \mathcal{M}(\nu,z^2) = \int_{-1}^{1} dx P(x,z^2) e^{-ix\nu}$



Reduced pseudo ITD, to avoid UV divergences

$$\mathfrak{M}(\nu,\mu^2) = \frac{\mathcal{M}(\nu,z^2)}{\mathcal{M}(0,z^2)} = \int_{0}^{1} \cos(\nu x) P(x,z^2)$$

Inverse problem if we want to determine $P(x,z^2)$

General description of the (my) problem

Lattice Data (ν) ↔ PDF (x)

Machine Learning of Nonlinear Partial Differential Equations
arXiv: 1708.00588

- Fortunately, I have an operator that relates 2 different spaces:

$$\mathcal{M}(\nu) = \mathcal{L}_\nu \mathcal{P}(x), \quad \text{where} \quad \mathcal{L}_\nu = \int_0^1 dx \cos(\nu x) \left(\quad \right).$$

- Can we determine $\mathcal{L}_\nu^{-1}$? if not you have a inverse problem.

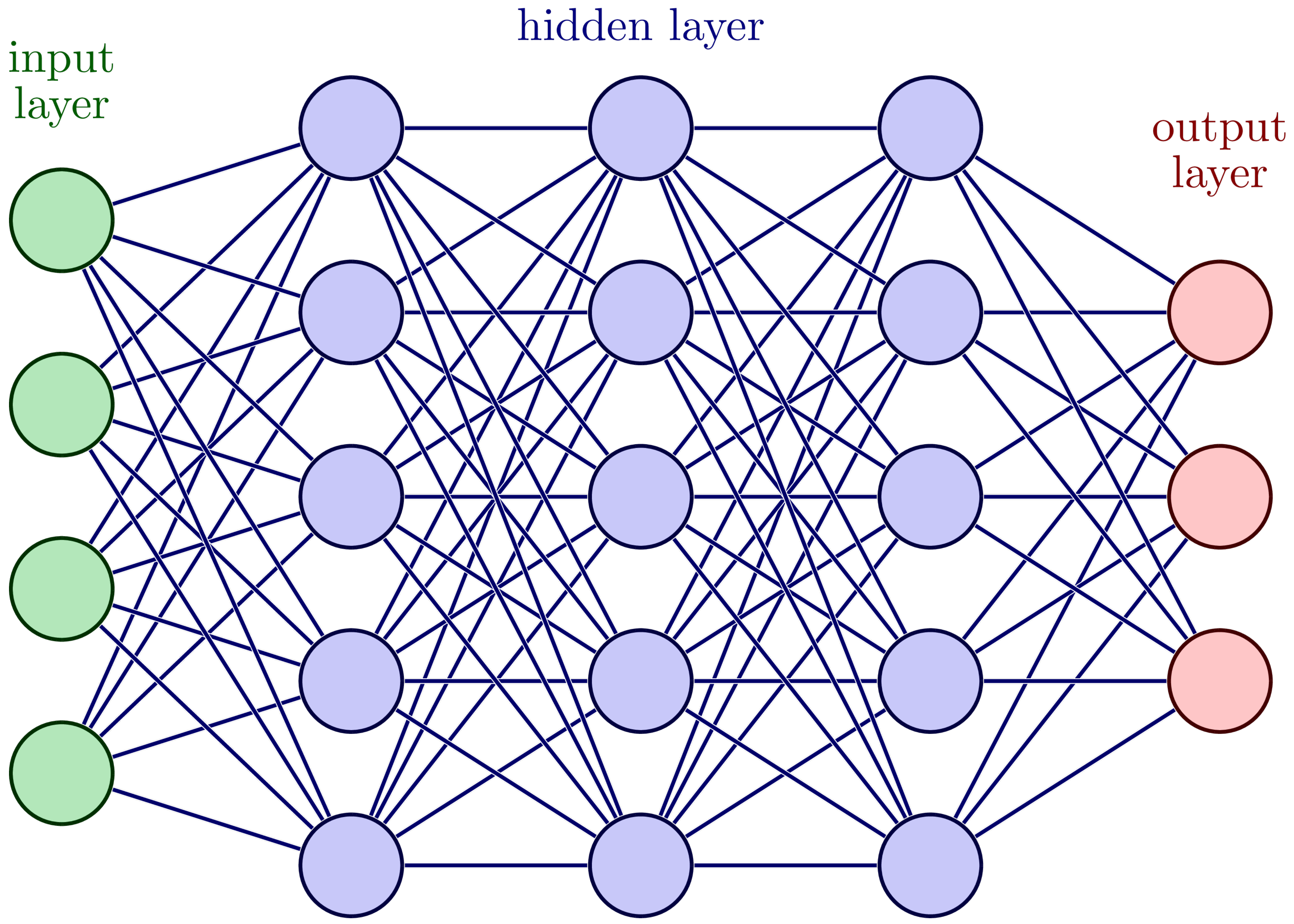
$$\mathfrak{M}_l = \Sigma_{lk}^{\perp} P_k$$

But if I have 12 data points in M
Can I only infer 12 data points of P(x)?

(Invertible) Neural Networks

[arXiv:1808.04730]
Analyzing Inverse Problems with Invertible Neural Networks

A brief introduction to NN and INN and MMD…in 1 min



- The architecture of a neural network is defined by the problem that we want to solve.

How can we define a INN?

Data + Affine Couplings (Invertible Mappings)

+ Activation functions + hyperparameters

ReLu

hidden layers

+ Loss Function + Optimizer = INN

Adam

I will focus on a brief description of these 3 main components

(Invertible) Neural Networks

A brief introduction to NN and INN and MMD…in 2 min

[arXiv:1605.08803]
Density estimation using Real NVP

- Can I create an invertible mapping?

How can we define a INN?

Data + Affine Couplings (Invertible Mappings)

+ Activation functions + hyperparameters
  ReLu — # hidden layers

+ Loss Function + Optimizer = INN
  Adam

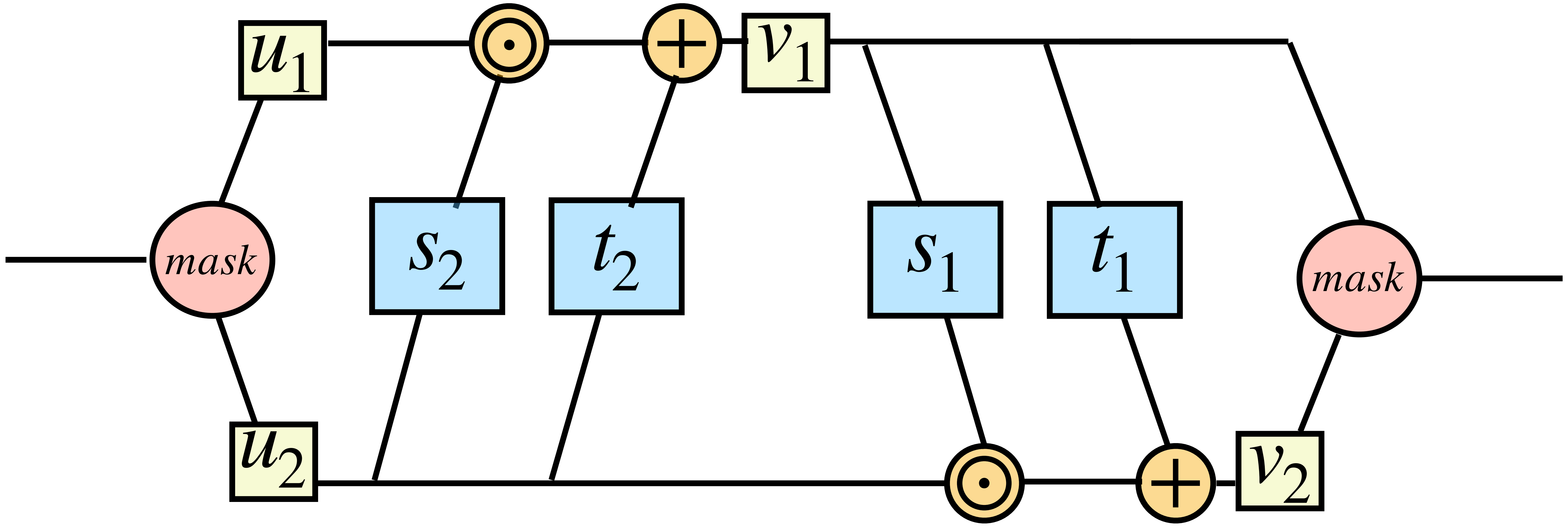
I will try to describe these 3 main components

$$v_1 = u_1 \odot e^{s_2(u_2)} + t_2(u_2)$$
$$v_2 = u_2 \odot e^{s_1(v_1)} + t_1(v_1)$$

Forward process

$$u_1 = e^{-s_2(u_2)} \odot (v_1 - t_2(u_2))$$
$$u_2 = e^{-s_1(v_1)} \odot (v_2 - t_1(v_1))$$

Backward process



$$x \odot y = (x_1 \cdot y_1, x_2 \cdot y_2)$$

(Invertible) Neural Networks

[arXiv:1605.08803]
Density estimation using Real NVP

A brief introduction to NN and INN and MMD…in 3 min

- Maximum Mean Discrepancy
- Any supervised loss

How can we define a INN?

Data + Affine Couplings (Invertible Mappings)

+ Activation functions + hyperparameters

ReLu — # hidden layers

+ Loss Function + Optimizer = INN

Adam

I will try to describe these 3 main components

$$\mathscr{L}_{Total} = aL_1 + bL_{MMD}(input) + cL_{MMD}(output)$$

$$L_{MMD}(x, y) = \frac{1}{n(n-1)} \sum_{i \neq j} k(x_i, x_j) + \frac{1}{n(n-1)} \sum_{i \neq j} k(y_i, y_j) - \frac{2}{n^2} \sum_{i,j} k(x_i, y_j)$$

MMD help us to preserve statistics

This loss helps with the regression process

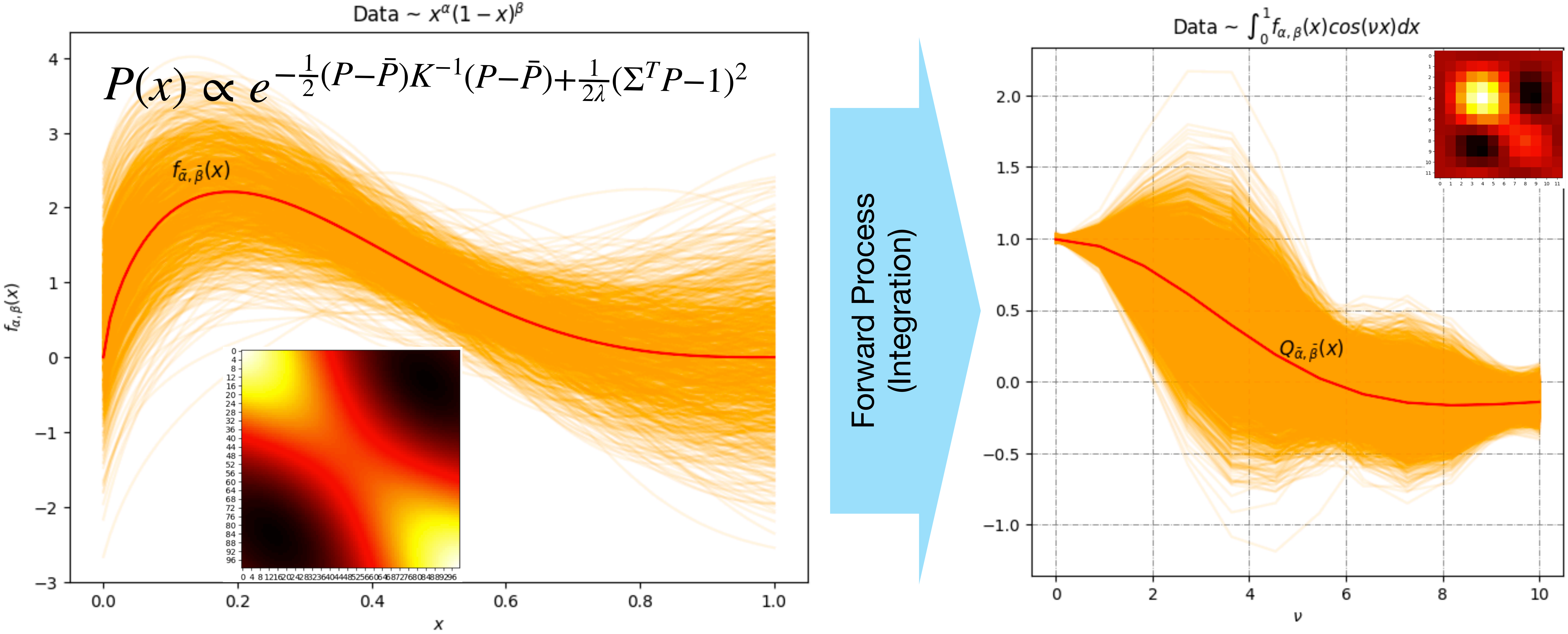
$$L_1(x, y) = |x - y|$$

Kernel

$$k(x, y) = \frac{1}{1 + \frac{|x-y|^2}{h}}$$

Data (Continuous functions in x)

How can I generate an significant amount of data?

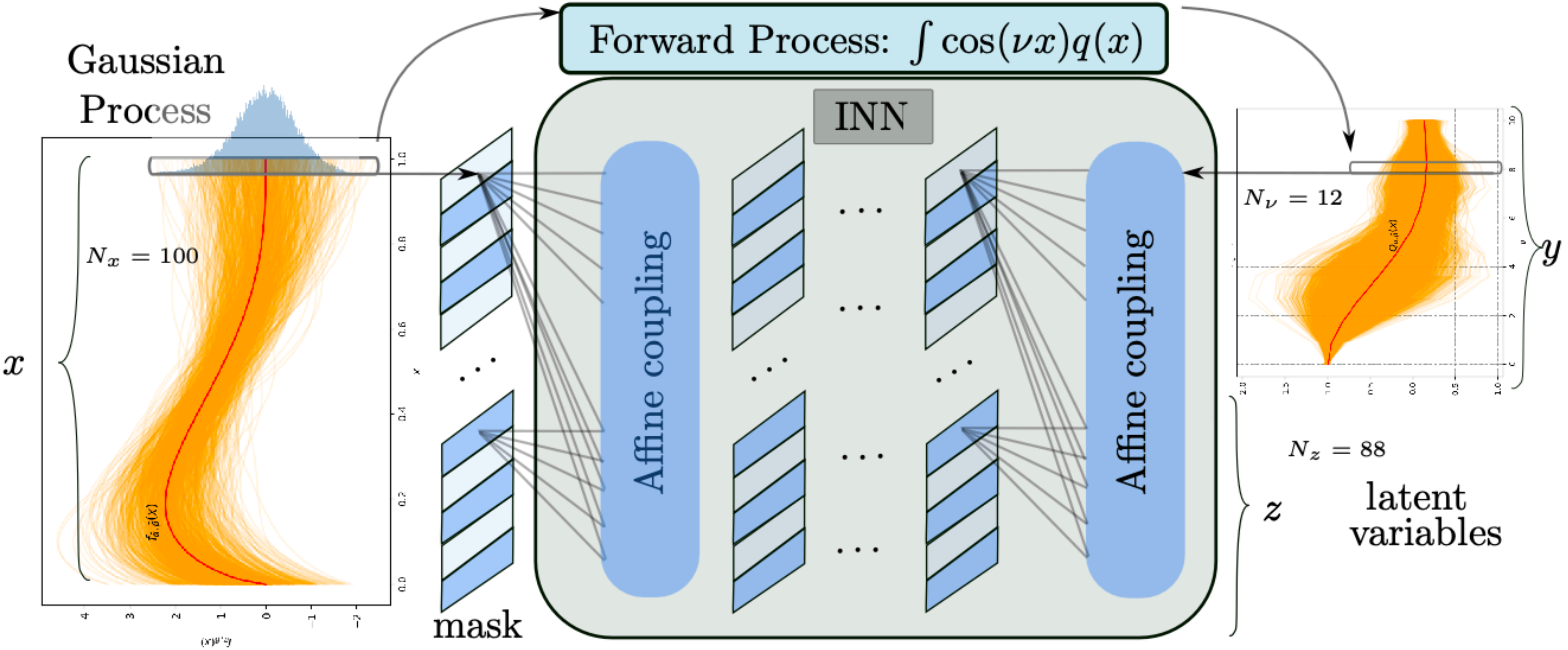


$$\alpha = 0.7 \qquad \beta = 3$$

INNs architecture

Latent variables and discretization

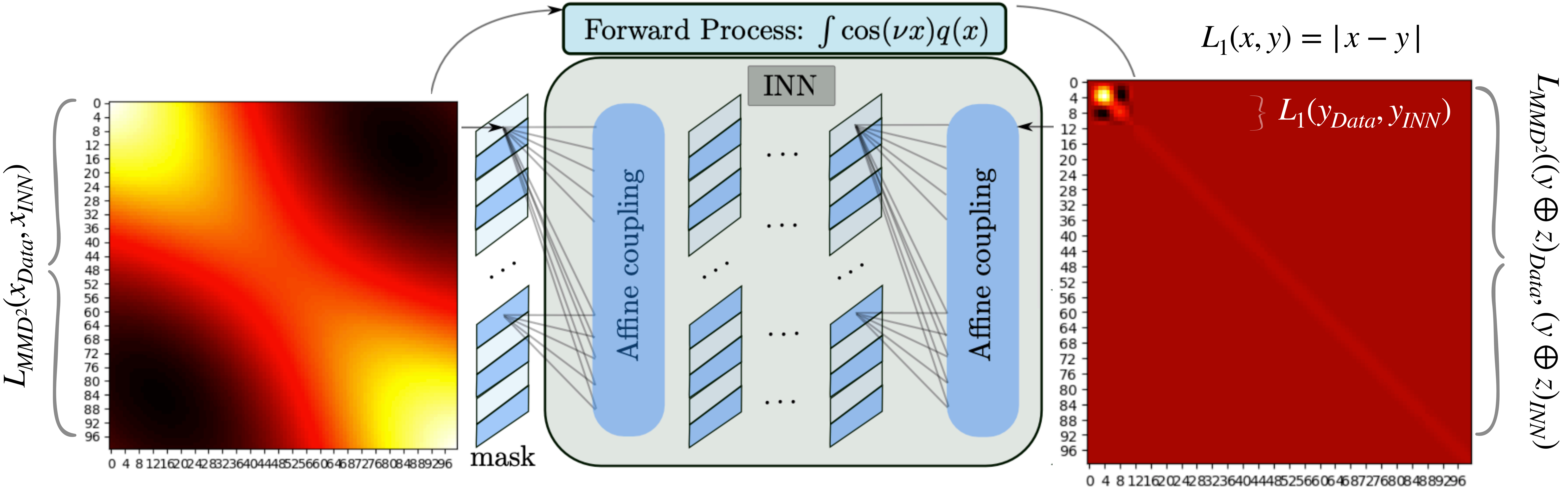
[arXiv:1808.04730]
Analyzing Inverse Problems with Invertible Neural Networks



Loss functions

Maximum Mean Discrepancy, and L1

[arXiv:1808.04730]
Analyzing Inverse Problems with Invertible Neural Networks



$$L_{MMD^2}(x, y) = \frac{1}{n(n-1)} \sum_{i \neq j} k(x_i, x_j) + \frac{1}{n(n-1)} \sum_{i \neq j} k(y_i, y_j) - \frac{2}{n^2} \sum_{i,j} k(x_i, y_j)$$

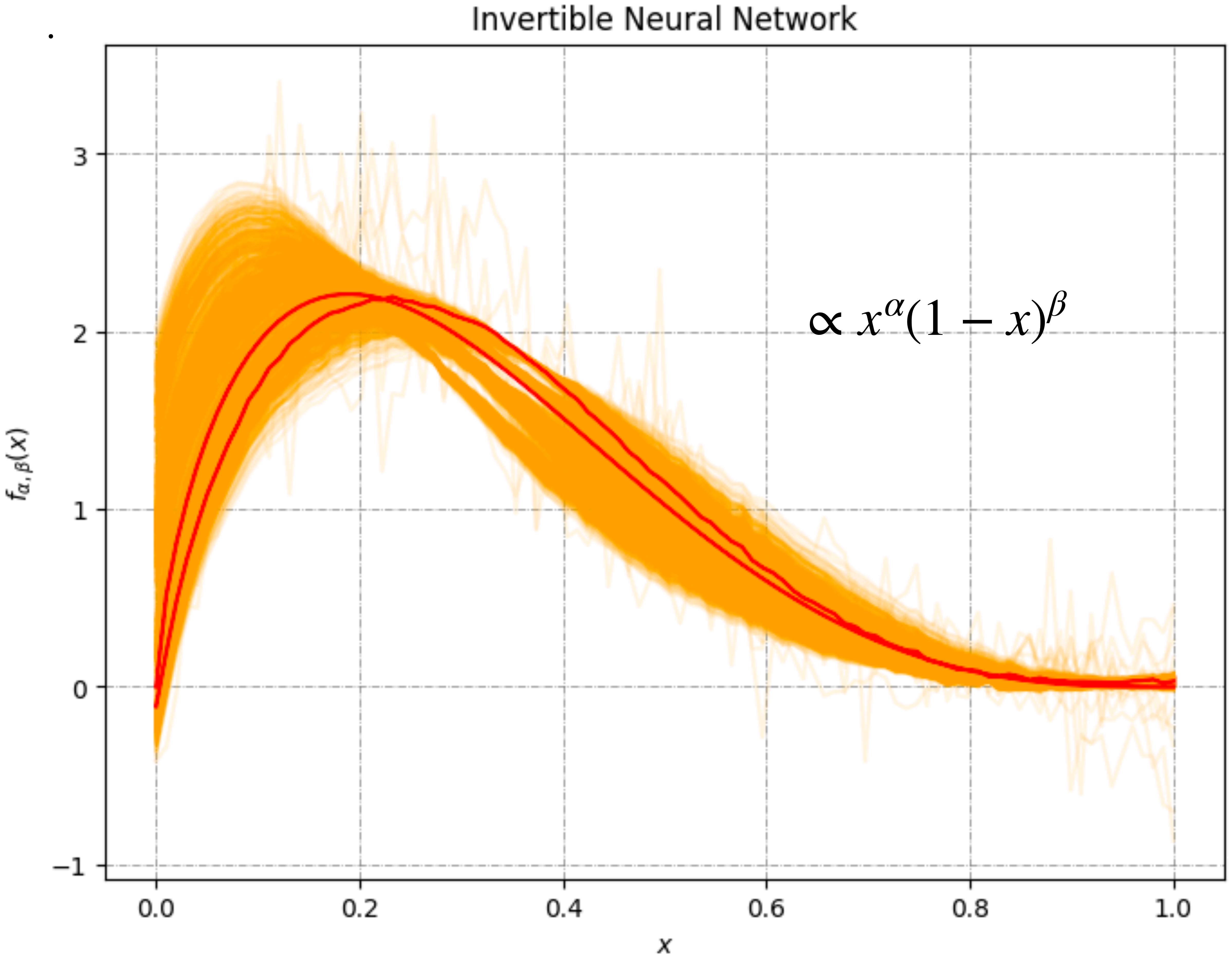
$$k(x, y) = \frac{1}{1 + \frac{|x-y|^2}{h}}$$

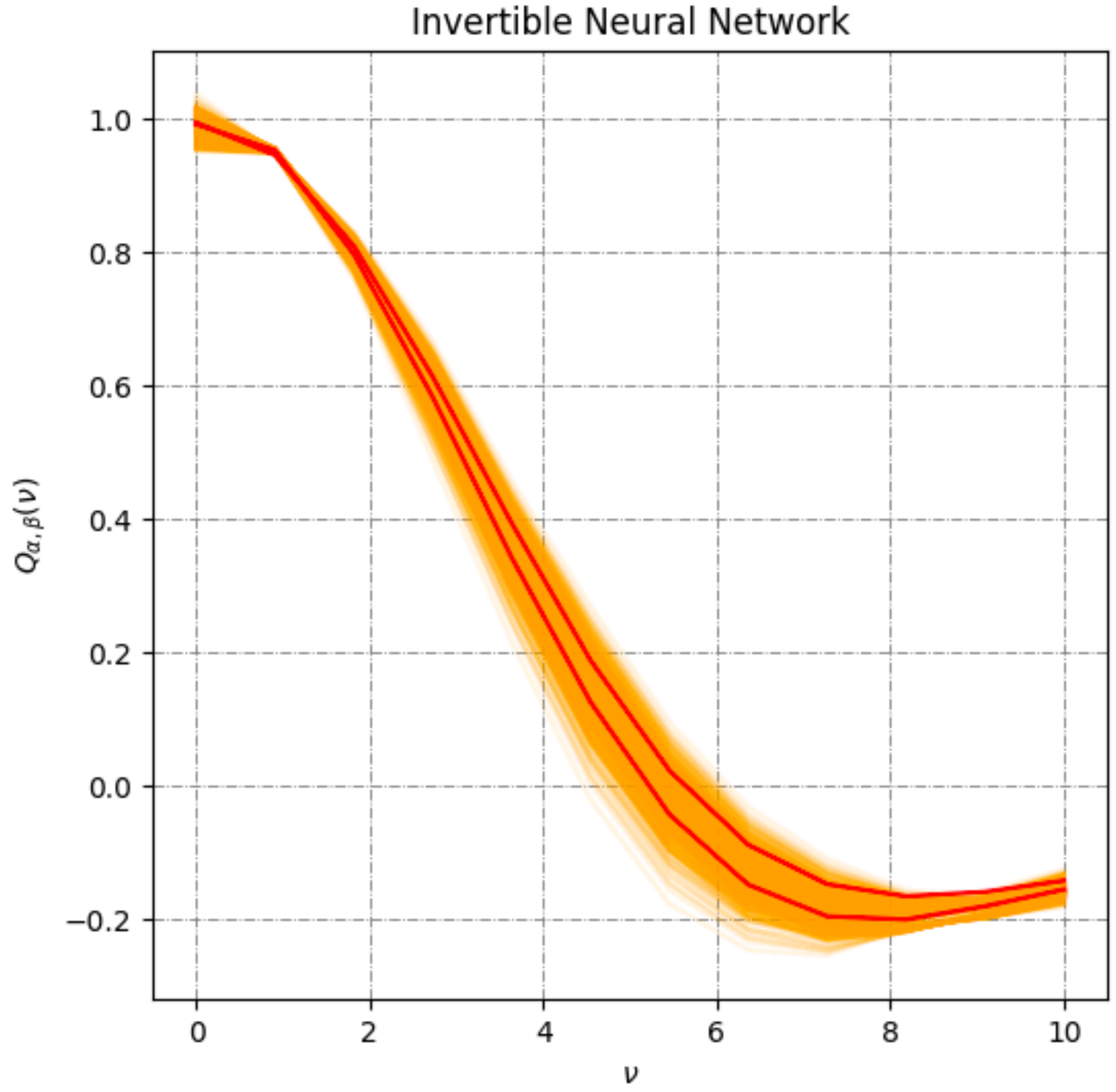
Results

Test data

- Data generated by other sampling process.

$\alpha = 0.7 \qquad \beta = 3$



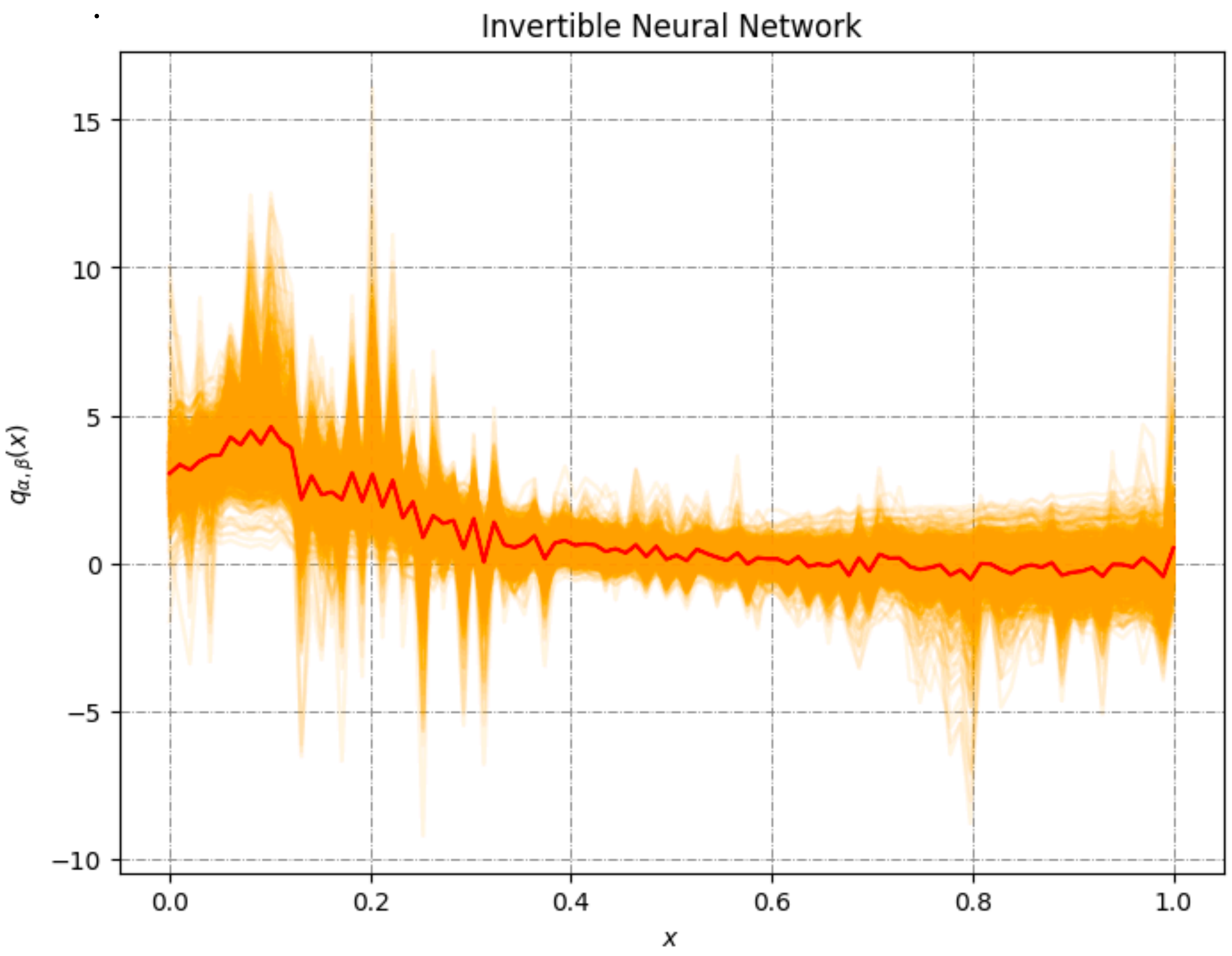


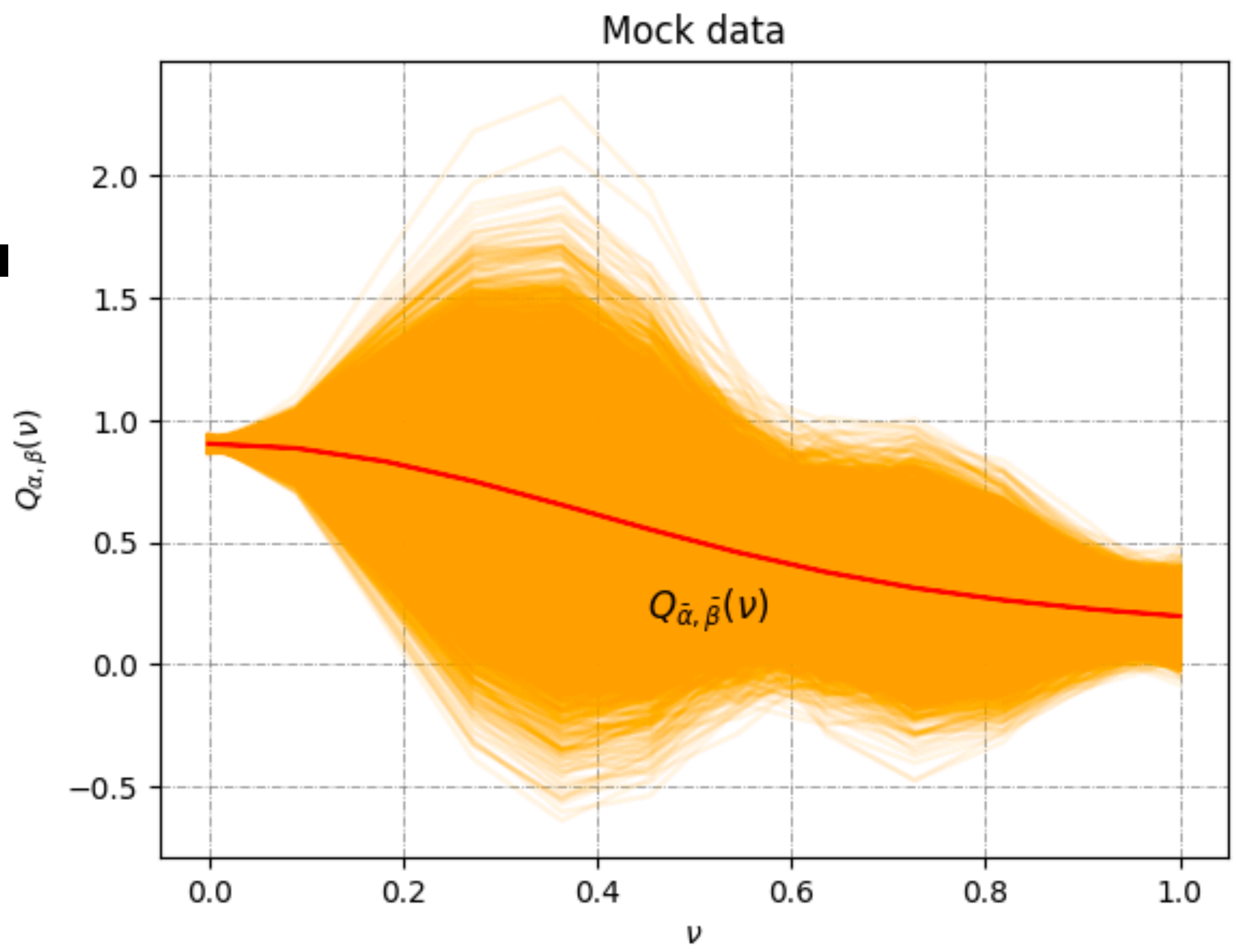
Results

Mock data

- Still working on it…

$\alpha = -0.3 \qquad \beta = 3$



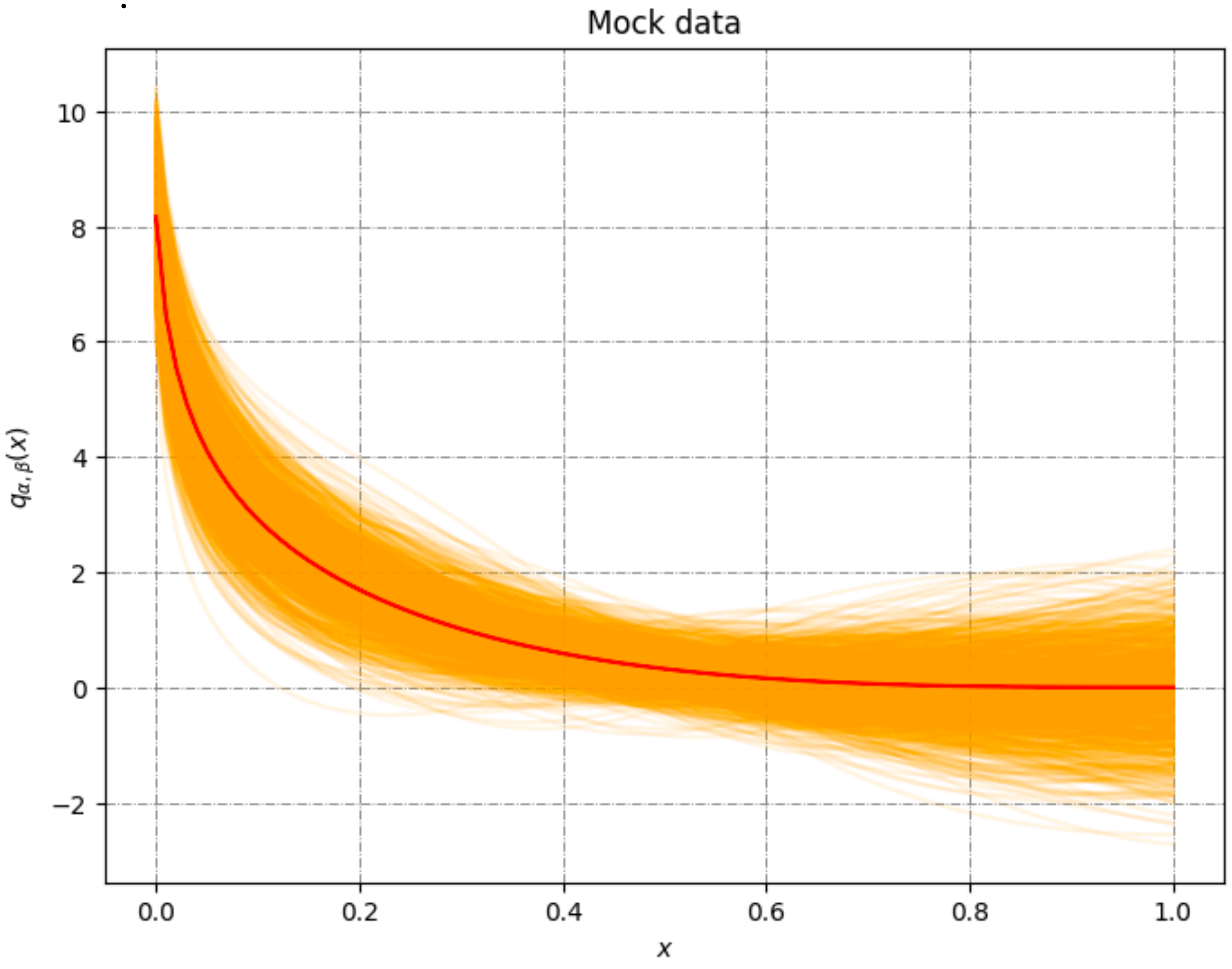


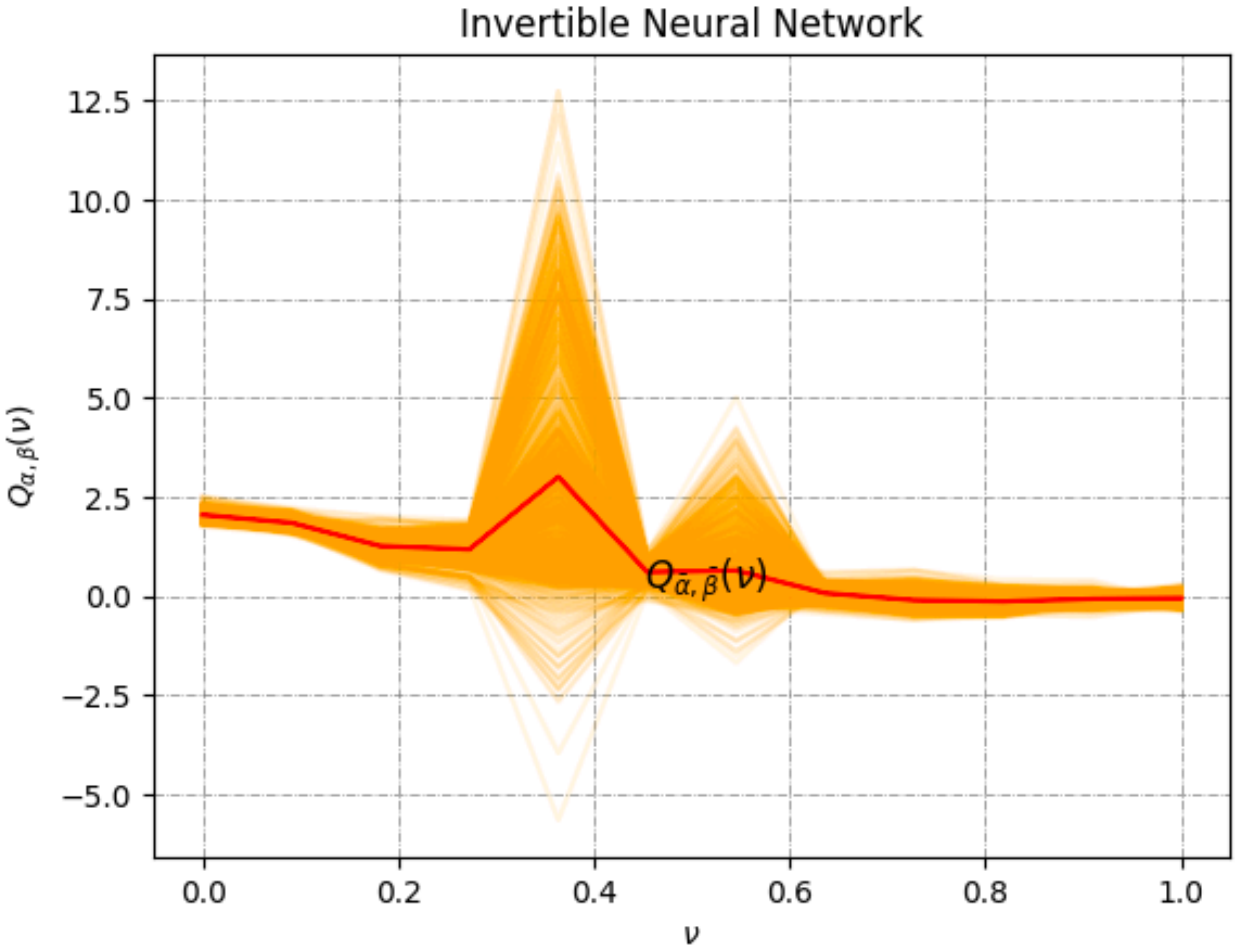
Results

Mock data

- Still working on it…

$\alpha = -0.3$ $\beta = 3$



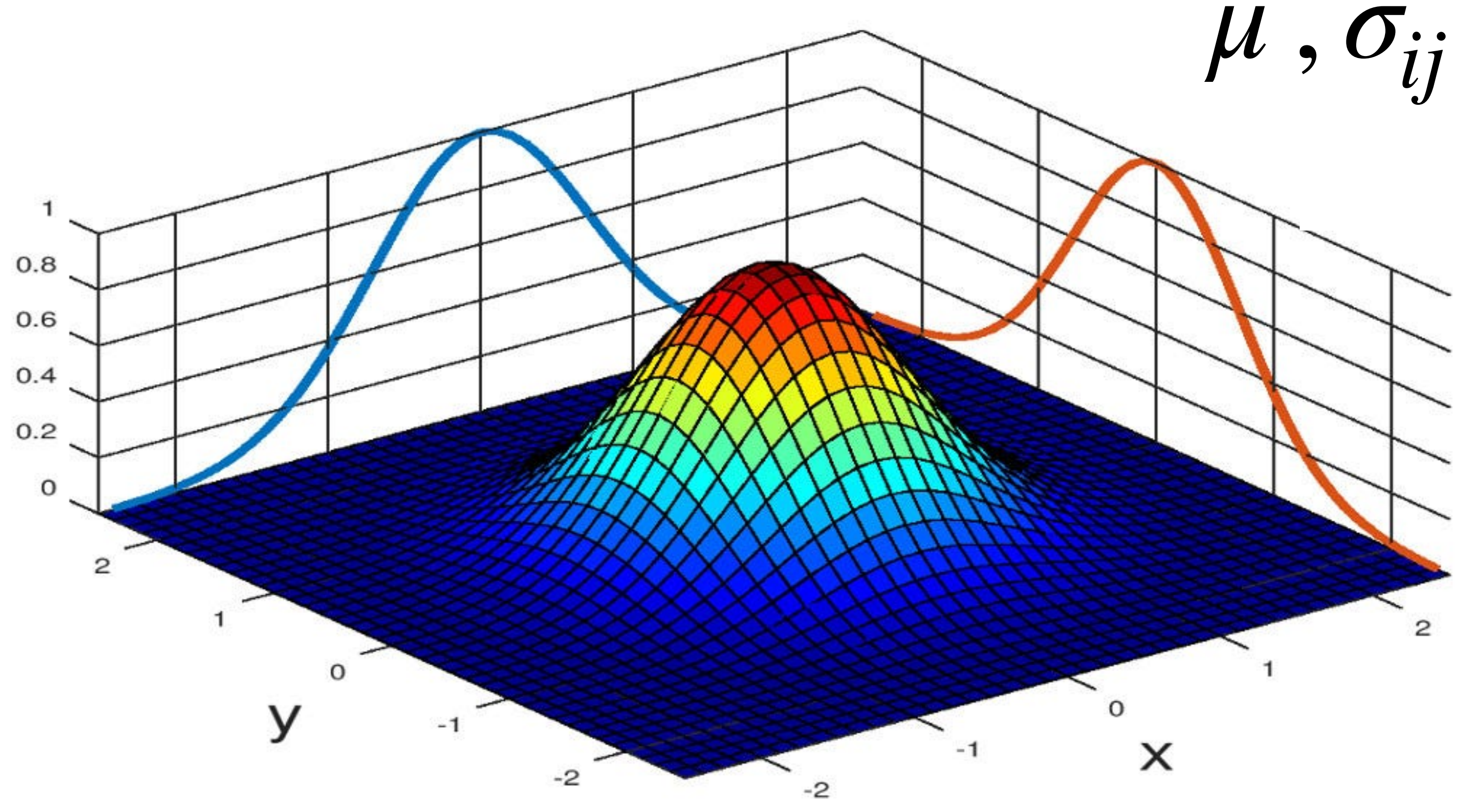


Gaussian process definitions

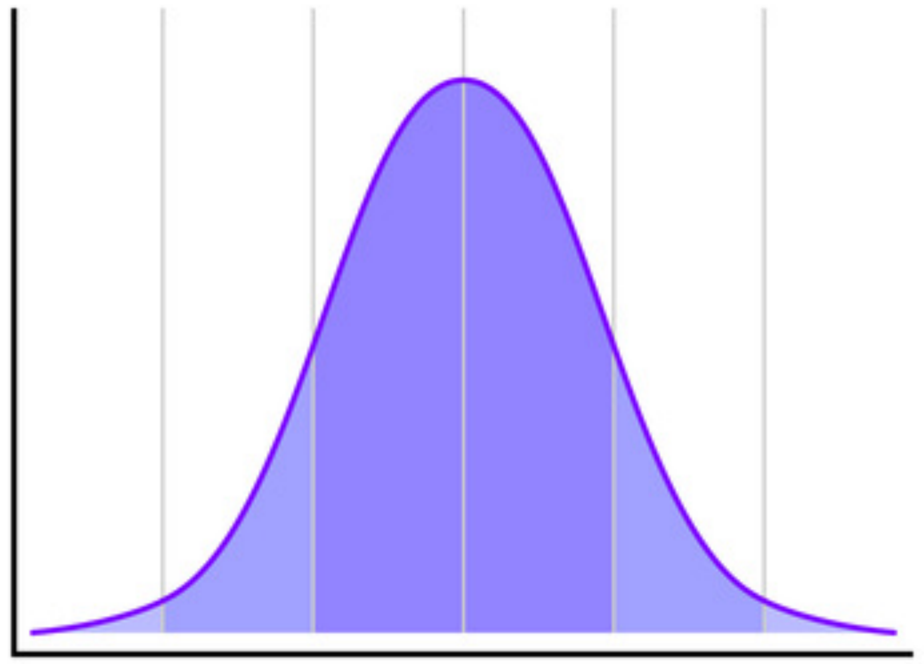
Stochastic Process

Rammunsen's book

$\overrightarrow{\mu}, \sigma_{ij}$



μ, σ



Generalization

$\mu(x), \sigma(x, x')$

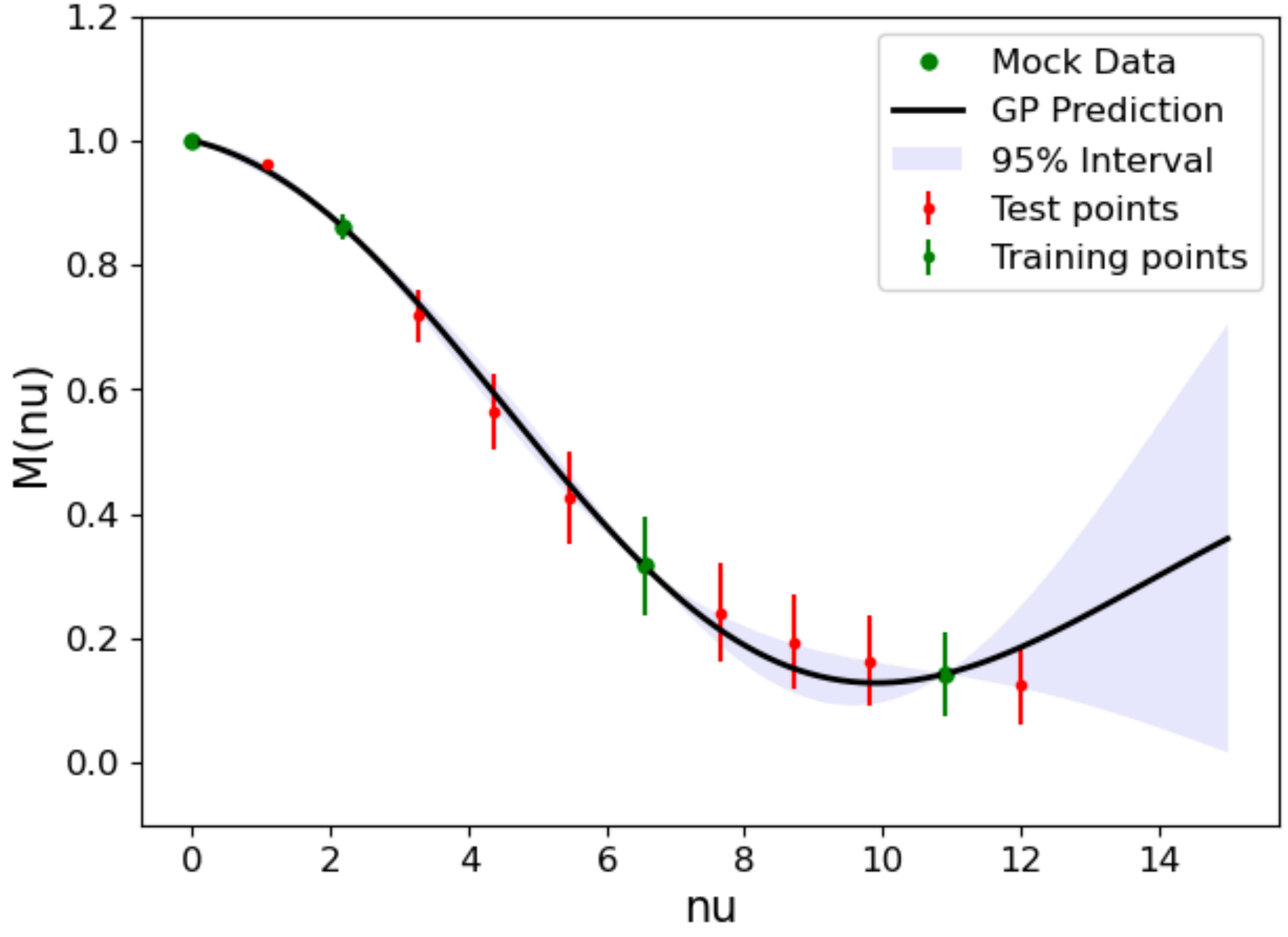
Try to imagine an infinite dimensional gaussian distribution

Bayes theorem

Lattice Data (ν) $\leftrightarrow$ PDF (x)

- Naively one may be tempted to write:

$$P(\mathcal{M}(\nu)|\{\mathcal{M}^l\},\theta,\mathcal{H}_i) = \frac{P(\{\mathcal{M}^l\}|\mathcal{M}(\nu),\theta,\mathcal{H}_i)P(\mathcal{M}(\nu)|\theta,\mathcal{H}_i)}{P(\{\mathcal{M}^l\}|\theta,\mathcal{H}_i)}.$$



Not useful for this problem

Bayes theorem

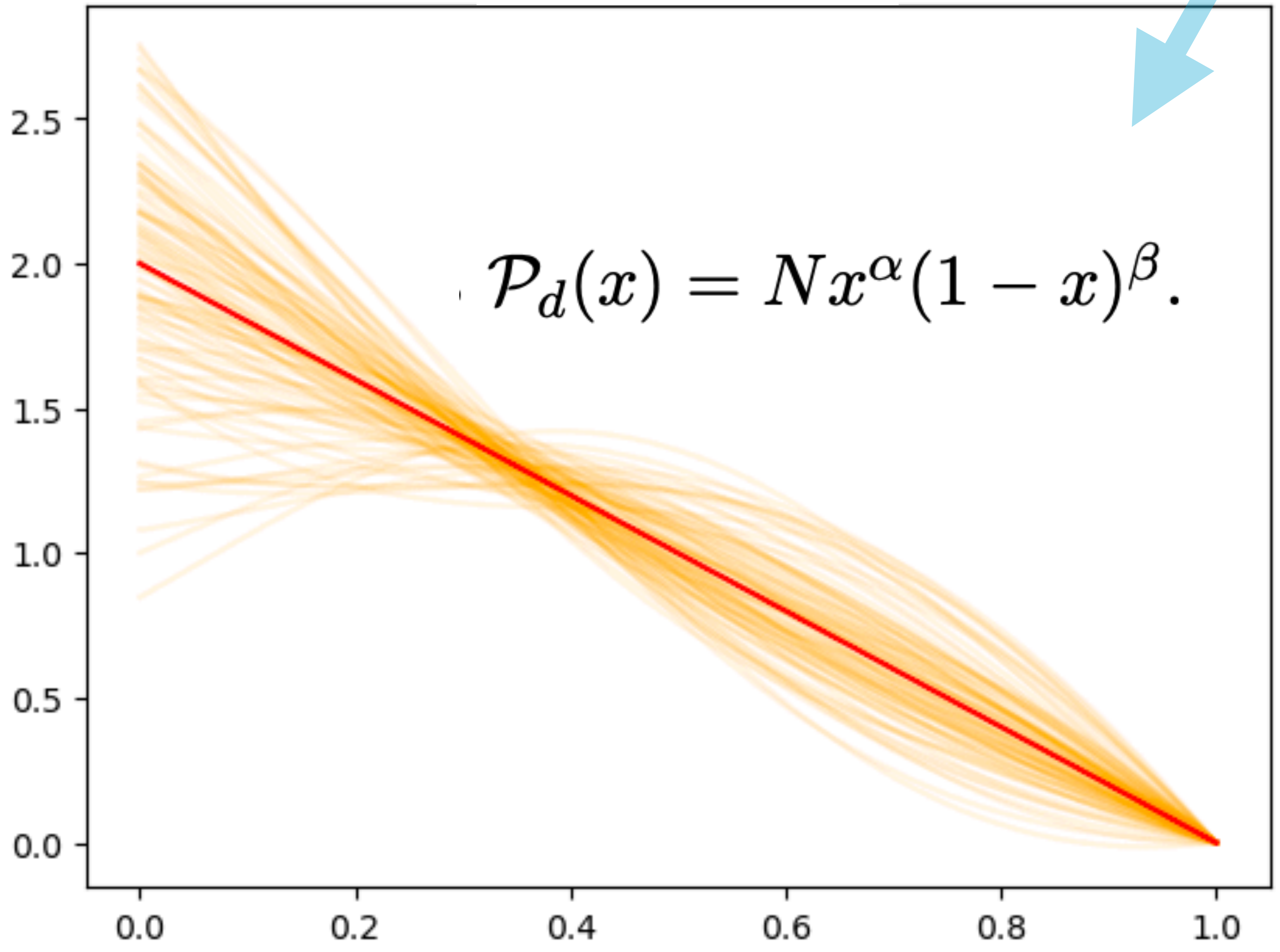
Lattice Data (ν) $\leftrightarrow$ PDF (x)

Machine Learning of Nonlinear Partial Differential Equations
arXiv: 1708.00588

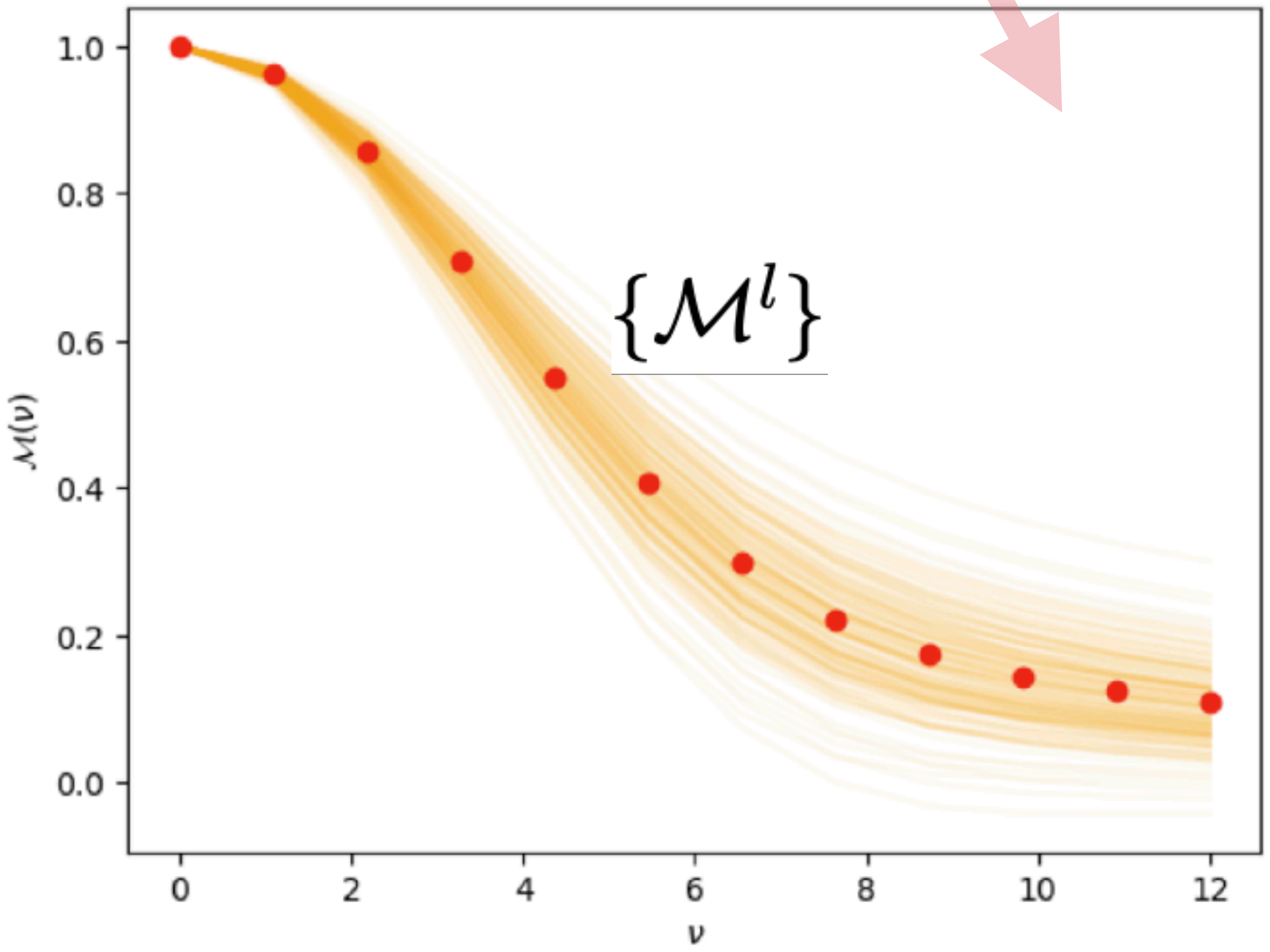
- Yamil, remember you said 2 DIFFERENT spaces, so you should write

$$P(\mathcal{P}(x)|\{\mathcal{M}^l\},\theta,\mathcal{H}_i) = \frac{P(\{\mathcal{M}^l\}|\mathcal{P}(x),\theta,\mathcal{H}_i)P(\mathcal{P}(x)|\theta,\mathcal{H}_i)}{P(\{\mathcal{M}^l\}|\theta,\mathcal{H}_i)}$$

Prior



Likelihood



The most important slide of my life…

Normal distribution

- Functional dependence on x

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

- Do you know this trick?

$$\frac{\partial \log(f)}{\partial x} = 0 \implies -\frac{1}{2}\frac{2(x-\mu)}{\sigma^2} = 0 \implies x = \mu$$

$$\left.\frac{\partial^2 \log(f)}{\partial^2 x}\right|_{x=\mu} = \sigma^{-2}$$

Generalize our results to GP

How my prior and likelihood looks like?

Reconstructing parton distribution functions from Ioffe time data: from Bayesian methods to neural networks, 10.1007/ jhep04(2019)057

Prior

$$e^{S_{prior}(\mathcal{P})} = P(\mathcal{P}(x)|\theta) = N_{prior} e^{-\frac{1}{2}\int_0^1 dx dx' [\mathcal{P}(x)-\mathcal{P}_d(x)] K^{-1}(x,x')[\mathcal{P}(x')-\mathcal{P}_d(x')]}$$

$$P_{constraint} = e^{-\frac{1}{2\lambda}\left(\int_0^1 dx \mathcal{P}(x)-1\right)^2 - \frac{1}{2\lambda_c}\left(\int_0^1 dx \mathcal{P}(x)\delta(1-x)\right)^2}.$$

Likelihood

$$e^{S_l(\mathcal{P})} = P(\{\mathcal{M}^l\}|\mathcal{P}(x),\theta,\mathcal{H}_i) = N_{likelihood} e^{-\frac{1}{2}[\mathcal{M}_i - \mathcal{L}_\nu \mathcal{P}(x)] C_{ij}^{-1} [\mathcal{M}_j - \int_0^1 dx \cos(\nu_j x)\mathcal{P}(x')]}$$

My job now is to calculate the posterior

$$\frac{\delta \log\left(P(\mathcal{P}(x)|\{\mathcal{M}^l\},\theta,\mathcal{H}_i)\right)}{\delta \mathcal{P}(x)} = 0 \implies \text{defines} \quad \bar{\mathcal{P}}(x)$$

and $$K_{post}^{-1}(x,x') = \frac{\delta^2 \log\left(P(\mathcal{P}(x)|\{\mathcal{M}^l\},\theta,\mathcal{H}_i)\right)}{\delta \mathcal{P}(x)\mathcal{P}(x')}\Big|_{\mathcal{P}(x)=\bar{\mathcal{P}}(x)}$$

Levels of inference in GP

How can we determine the hyperparameters?

Rammunsen's book

- 1st level (continuous variables P(x) and K(x.x')).

$$P(\mathcal{P}(x)|\{\mathcal{M}^l\},\theta,\mathcal{H}_i) = \frac{P(\{\mathcal{M}^l\}|\mathcal{P}(x),\theta,\mathcal{H}_i)P(\mathcal{P}(x)|\theta,\mathcal{H}_i)}{P(\{\mathcal{M}^l\}|\theta,\mathcal{H}_i)}$$

- 2nd Level (Hyperparameters) basically how I control the parameters of the first level.

$$P(\theta|\{\mathcal{M}^l\},\mathcal{H}_i) = \frac{P(\{\mathcal{M}^l\}|\theta,\mathcal{H}_i)P(\theta|\mathcal{H}_i)}{P(\{\mathcal{M}^l\}|\mathcal{H}_i)}$$

- The 3rd level is used to evaluate the models and its performance.

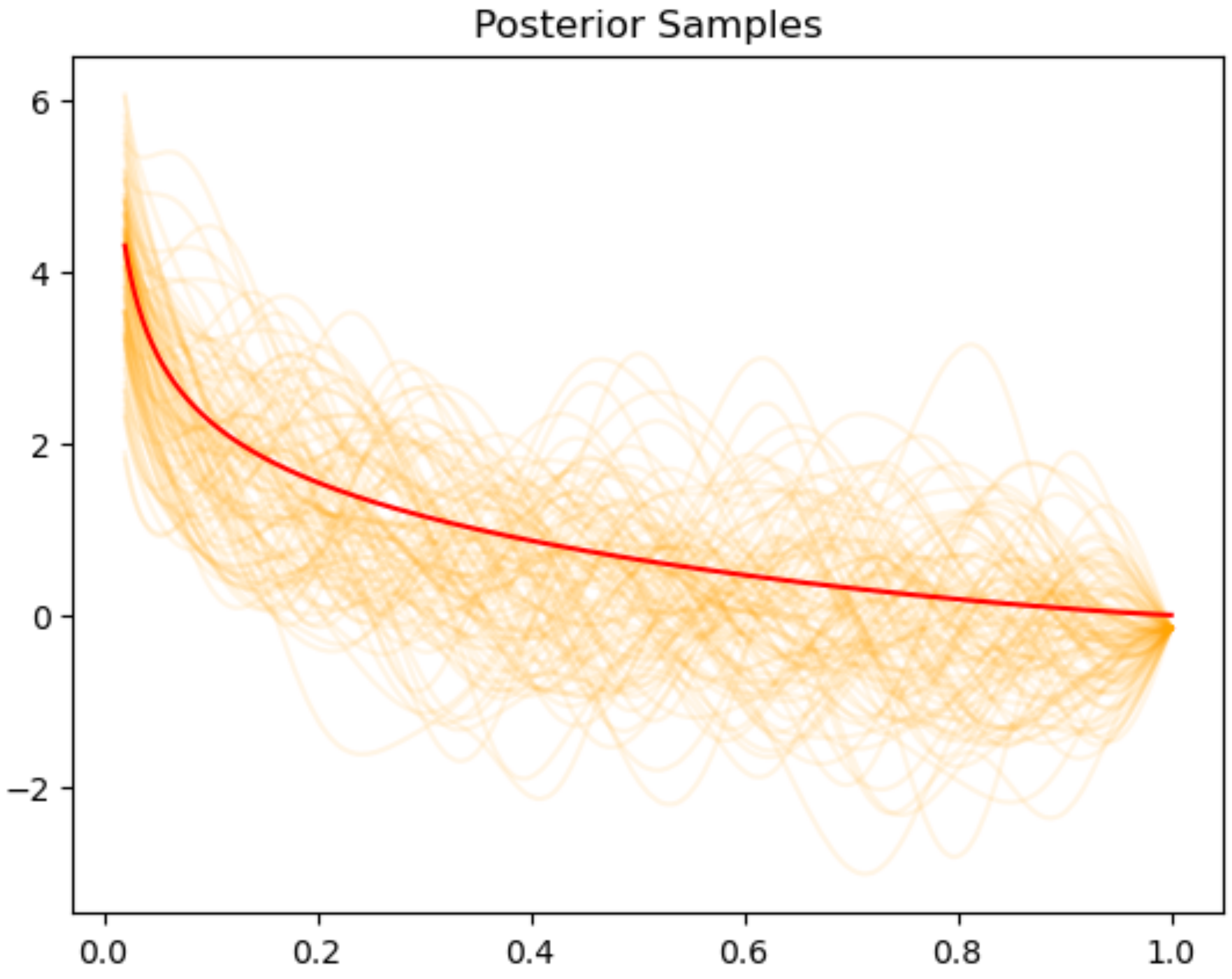
Preliminary results…

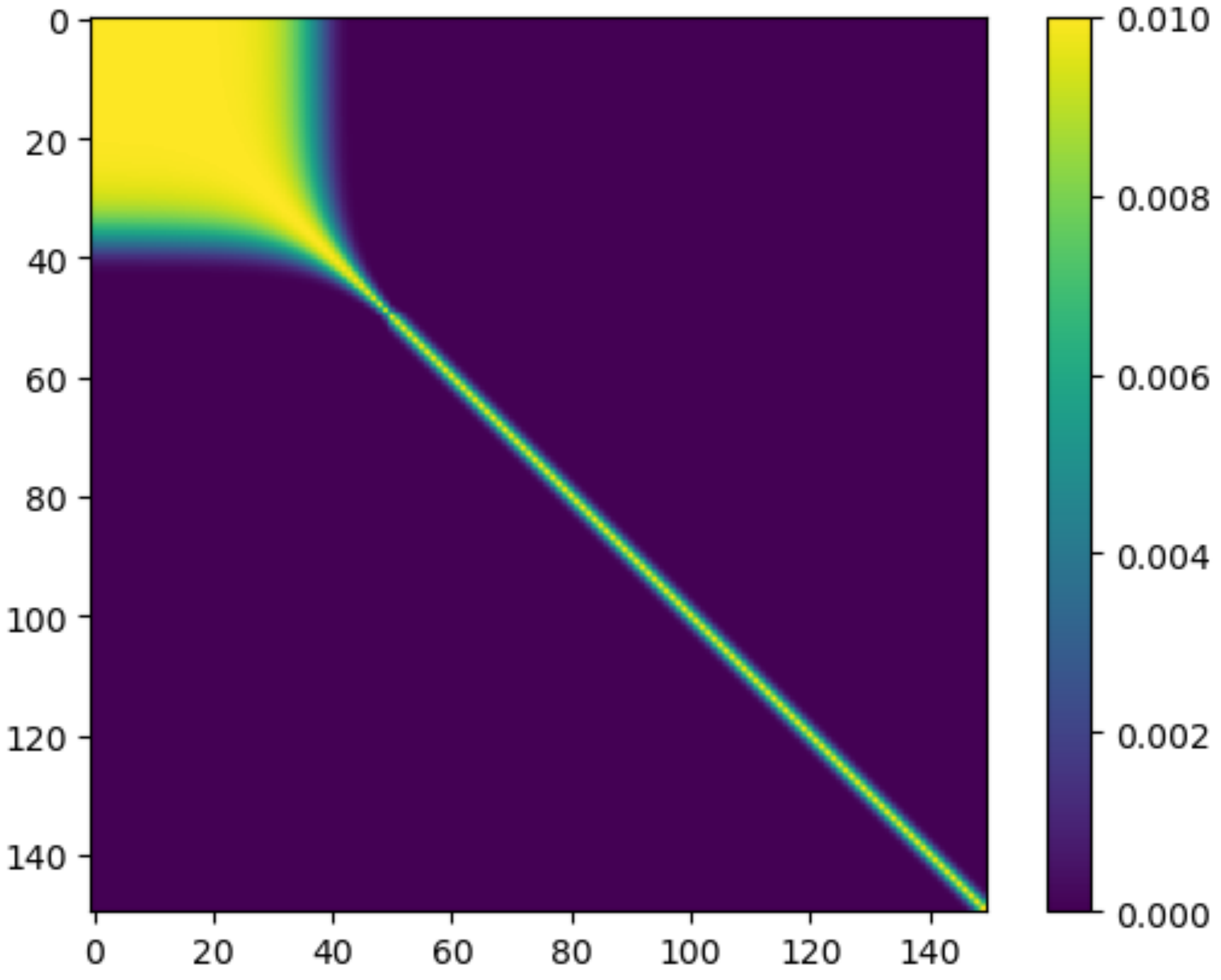
Hyperparameters

$l = 0.1$ $\sigma = 0.01$ $\alpha = -0.338$ $\beta = 1.195$

- Still working on it…

$$K_{rbf}(x, x') = \sigma^2 e^{\frac{(x-x')^2}{2l^2}}$$





Future work

Ways to improve this work…

- Sample hyper-parameters instead of minimize the negative log marginal likelihood

- Explore the posibility of use Hierarchical Models in the implementation and Pymc

- Include the evolution of z, which follow a similar evolution to the DGLAP equation.

- Extend this work to GPDs!!

References

Papers and Books

- Maziar Raissi and George E. Karniadakis. “Hidden Physics Models: Machine Learning of Nonlinear Partial Differential Equations”. In: CoRR abs/ 1708.00588 (2017). arXiv: 1708.00588. url: http://arxiv.org/abs/1708.00588.

- Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. “Machine learning of linear differential equations using Gaussian processes”. In: Journal of Computational Physics 348 (Nov. 2017), pp. 683–693. issn: 0021-9991. doi: 10.1016/j.jcp.2017.07. 050. url: http://dx.doi.org/10.1016/j.jcp.2017.07.050.

- Carl Edward Rasmussen and Christopher K. I. Williams. Gaussian processes for ma- chine learning. Adaptive computation and machine learning. MIT Press, 2006, pp. I– XVIII, 1–248. isbn: 026218253X.

References

Papers

- Lin, Huey-Wen, Nocera, Emanuele R., Olness, Fred, Orginos, Kostas, Rojo, Juan, et. al. Parton distributions and lattice QCD calculations: A community white paper. USA. https://doi.org/10.1016/j.ppnp.2018.01.007

- Joseph Karpie et al. “Reconstructing parton distribution functions from Ioffe time data: from Bayesian methods to neural networks”. In: Journal of High Energy Physics 2019.4 (Apr. 2019). doi: 10.1007/ jhep04(2019)057. url: https://doi.org/10.1007%2Fjhep04%282019%29057.

- Lynton Ardizzone et al. “Analyzing Inverse Problems with Invertible Neural Networks”. In: (2019). arXiv: 1808.04730 [cs.LG].

Back-up slides

Generalize our results to GP

Posterior

My job now is to calculate the posterior

$$\frac{\delta S_{post}}{\delta \mathcal{P}(x)} = (\mathcal{P} - \mathcal{P}_d)\, K^{-1}(x) + \frac{\mathcal{I}(x)(\mathcal{I} \circ \mathcal{P} - 1)}{\lambda} + \frac{\delta_1(x)(\delta_1 \circ \mathcal{P})}{\lambda_c} + B_i^{\perp}(x) C_{ij}^{-1} (B \circ \mathcal{P} - \mathcal{M})_j$$

$$\frac{\delta^2 S_{post}}{\delta \mathcal{P}(x) \delta \mathcal{P}(x)}\bigg|_{\mathcal{P}=\bar{\mathcal{P}}} = K^{-1}(x, x') + \frac{\mathcal{I}(x)\mathcal{I}(x')}{\lambda} + \frac{\delta_1(x)\delta_1(x)}{\lambda_c} + B_i^{\perp}(x) C_{ij}^{-1} B_j(x') \quad (2.13)$$

After a lot of algebra…

Generalize our results to GP

Posterior

Reconstructing parton distribution functions from Ioffe time data: from Bayesian methods to neural networks, 10.1007/ jhep04(2019)057

After a lot of algebra…

$$P(\mathcal{P}(x)|\{\mathcal{M}^l\},\theta,\mathcal{H}_i) = \frac{e^{-\left\{\frac{1}{2}S_{post}\left(\bar{\mathcal{P}}(x)\right)+\frac{1}{2}\int_0^1 dx'\left(\mathcal{P}(x)-\bar{\mathcal{P}}(x)\right)\frac{\delta^2 S_{post}}{\delta\mathcal{P}(x)\delta\mathcal{P}(x')}\Big|\left(\mathcal{P}(x')-\bar{\mathcal{P}}(x')\right)\right\}}}{P\left(\{\mathcal{M}_i\}\,|\theta,\mathcal{H}_i\right)}$$

$$= \frac{e^{-\frac{1}{2}\left(\mathcal{P}(x)-\bar{\mathcal{P}}(x)\right)K^{-1}_{post}(x,x')\left(\mathcal{P}(x')-\bar{\mathcal{P}}(x')\right)+S_{prior}(\bar{\mathcal{P}})+S_l(\bar{\mathcal{P}})}}{P\left(\{\mathcal{M}_i\}\,|\theta,\mathcal{H}_i\right)} \qquad (2.19)$$

Where

$$\bar{\mathcal{P}}(x) = \mathcal{P}_d(x) - (K_{post}\circ B_i^{\perp})(x)C_{ij}^{-1}\left(\mathcal{M}_j - B_j\circ\mathcal{P}_d\right)$$

Levels of inference

Lattice Data (ν) $\leftrightarrow$ PDF (x)

$$P(\mathcal{P}(x)|\{\mathcal{M}^l\},\theta,\mathcal{H}_i) = \frac{P(\{\mathcal{M}^l\}|\mathcal{P}(x),\theta,\mathcal{H}_i)P(\mathcal{P}(x)|\theta,\mathcal{H}_i)}{P(\{\mathcal{M}^l\}|\theta,\mathcal{H}_i)}$$

$$P(\theta|\{\mathcal{M}^l\},\mathcal{H}_i) = \frac{P(\{\mathcal{M}^l\}|\theta,\mathcal{H}_i)P(\theta|\mathcal{H}_i)}{P(\{\mathcal{M}^l\}|\mathcal{H}_i)}$$

$$\bar{\mathcal{P}}(x) = \mathcal{P}_d(x) - (K_{post}\circ B_i^{\perp})(x)C_{ij}^{-1}(\mathcal{M}_j - B_j\circ\mathcal{P}_d)$$

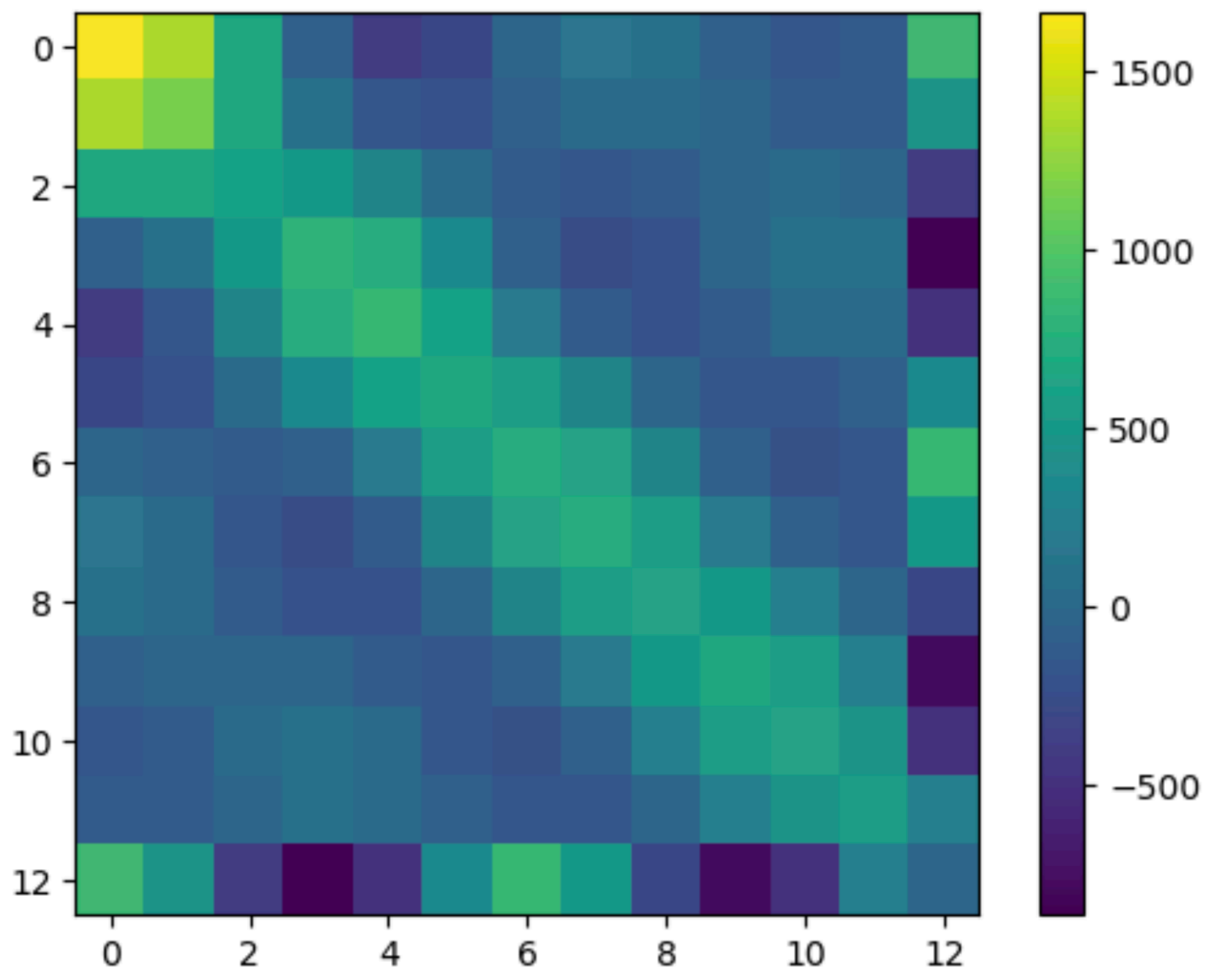
$$S_{Evidence} = \frac{1}{2}(\mathcal{M}_i - B_i\circ\mathcal{P}_d)\bar{C}_{ij}^{-1}(\mathcal{M}_i - B_i\circ\mathcal{P}_d) + \frac{1}{2}\log\det\left(\bar{C}_{ij}\right)$$

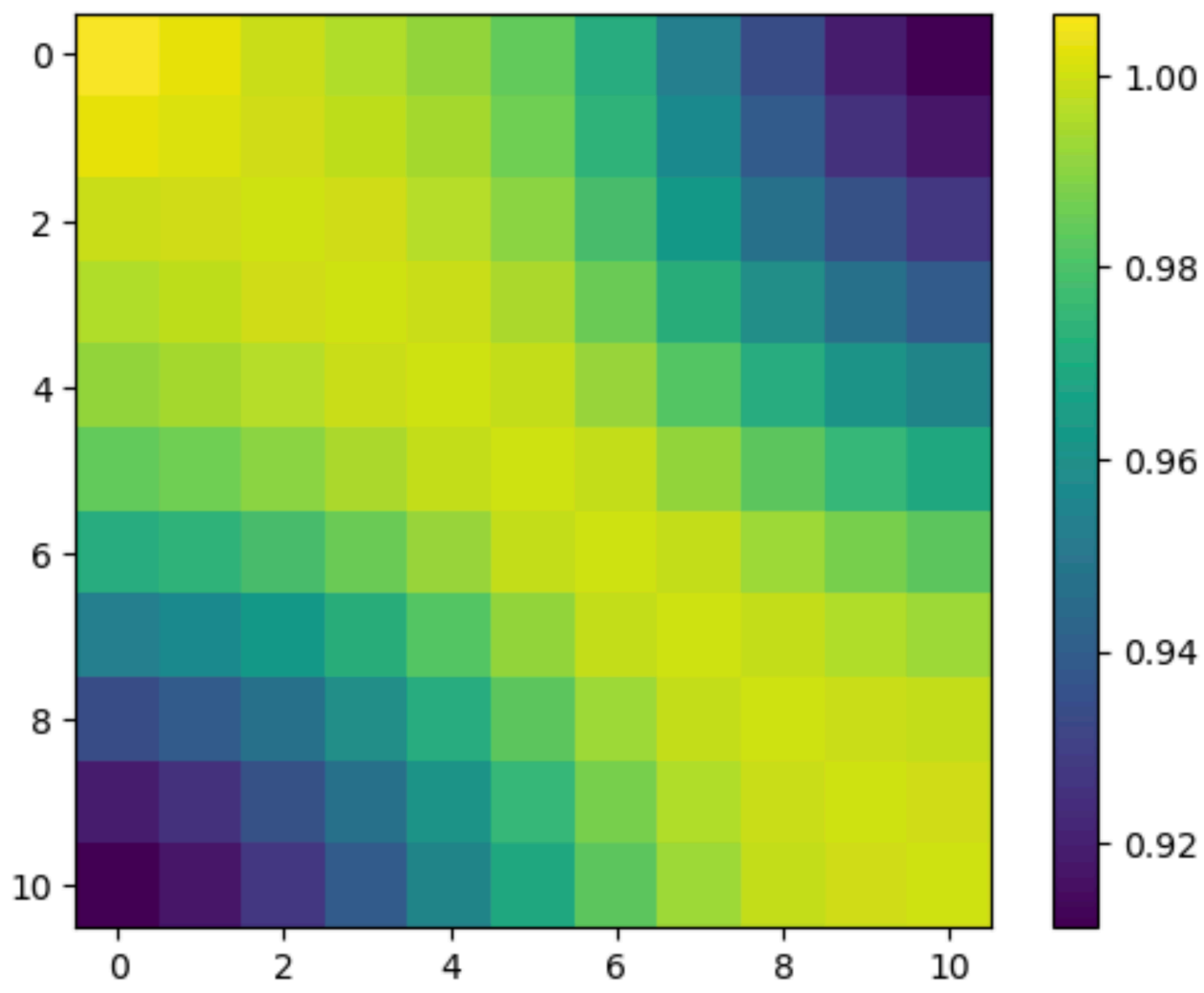
Final Result

Lattice Data (ν) $\leftrightarrow$ PDF (x)

Lets visualize

$$S_{Evidence} = \frac{1}{2}(\mathcal{M}_i - B_i \circ \mathcal{P}_d)\bar{C}_{ij}^{-1}(\mathcal{M}_i - B_i \circ \mathcal{P}_d) + \frac{1}{2}\log\det\left(\bar{C}_{ij}\right)$$



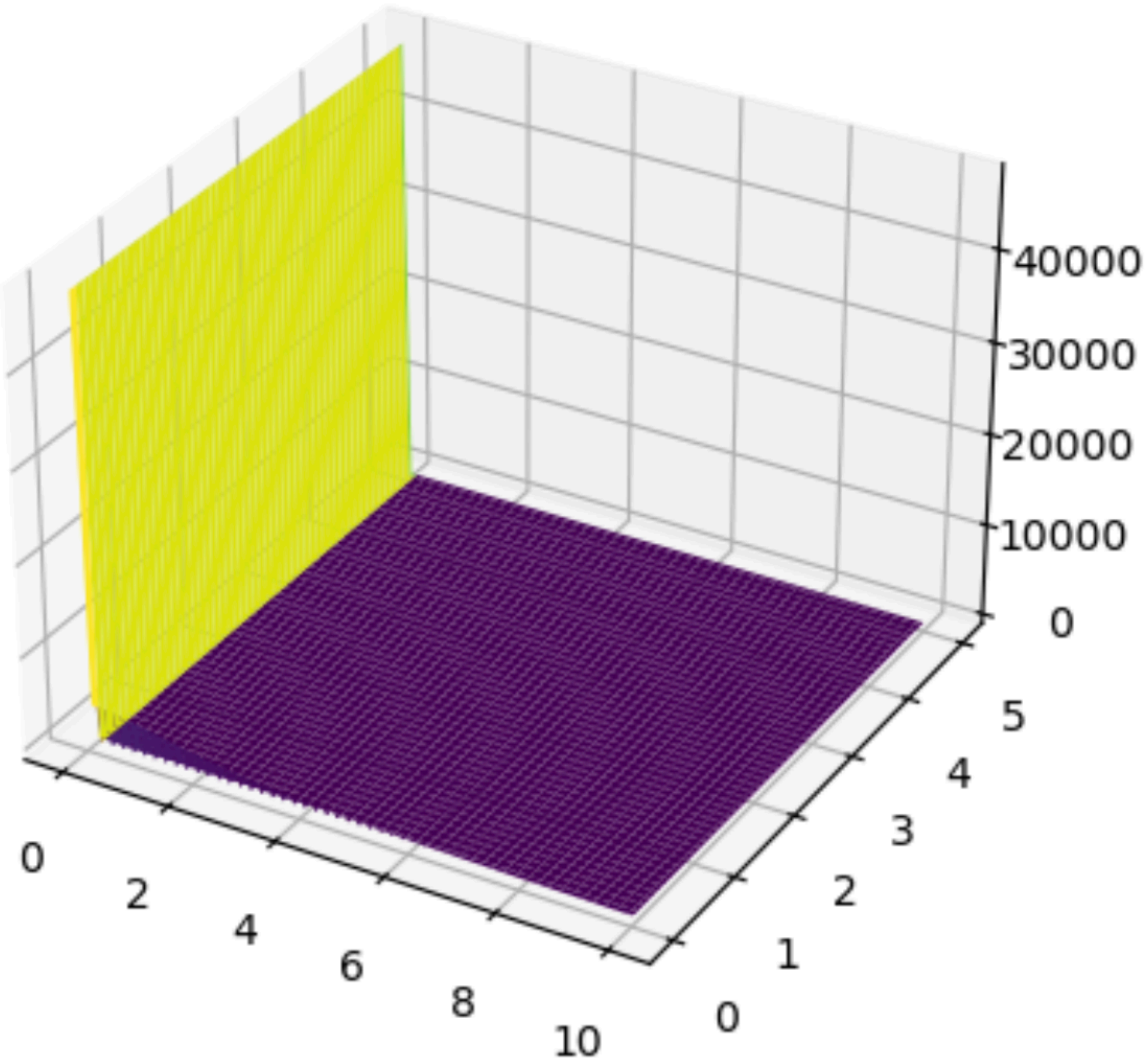


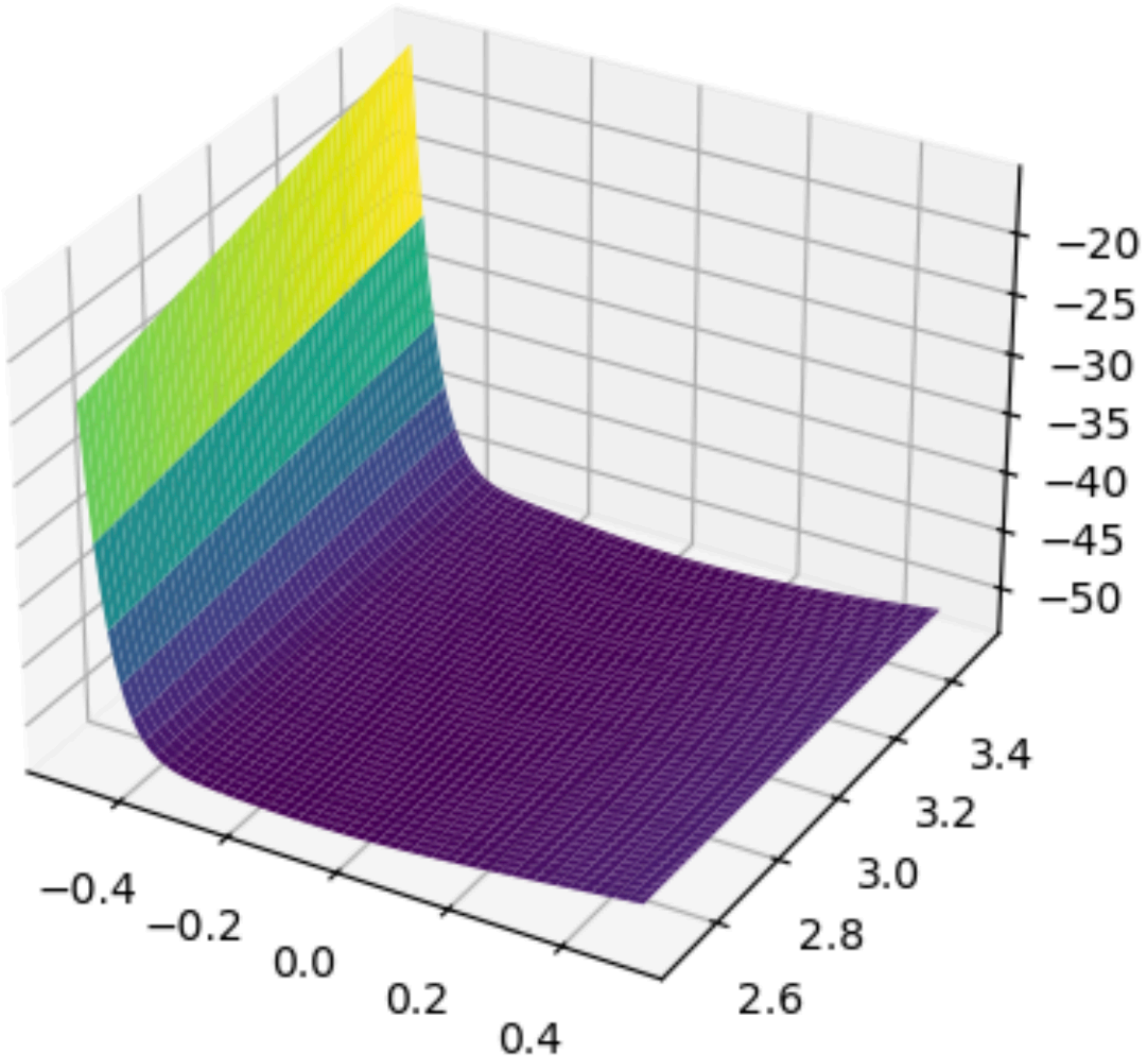
Evidence

Lattice Data (ν) $\leftrightarrow$ PDF (x)

Lets visualize

$$S_{Evidence} = \frac{1}{2}(\mathcal{M}_i - B_i \circ \mathcal{P}_d)\bar{C}_{ij}^{-1}(\mathcal{M}_i - B_i \circ \mathcal{P}_d) + \frac{1}{2}\log\det\left(\bar{C}_{ij}\right)$$





Parton distribution functions (PDFs)

A brief history of time blah…

- How can we describe proton's structure?

$e^- P^+ \to e^- X$

PDF

$$\sigma(e^- P^+ \to e^- X) = \sum_{i=partons} \int_0^1 dx f_i(x) \hat{\sigma}(e^- p_i \to e^- X)$$

Unpolarized PDFs Helicity-averaged

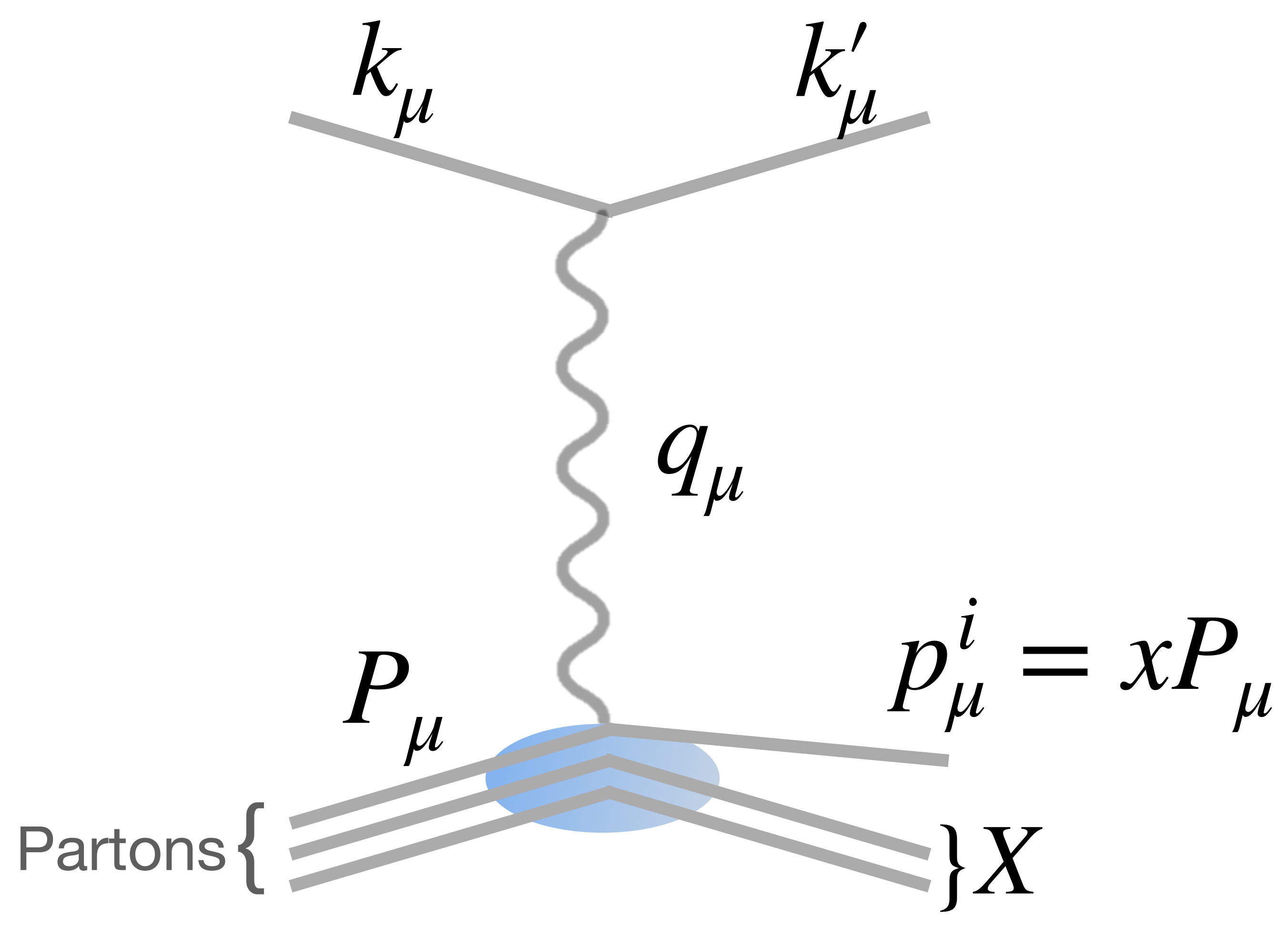
$$f_i(x, \mu^2) = f_i^{\rightarrow}(x, \mu^2) + f_i^{\leftarrow}(x, \mu^2)$$

We will focus on this

Polarized PDFs

$$\Delta f_i(x, \mu^2) = f_i^{\rightarrow}(x, \mu^2) - f_i^{\leftarrow}(x, \mu^2)$$

$f_i = \{g, u, \bar{u}, d, \bar{d}, s, \bar{s}, \ldots\} \longrightarrow$



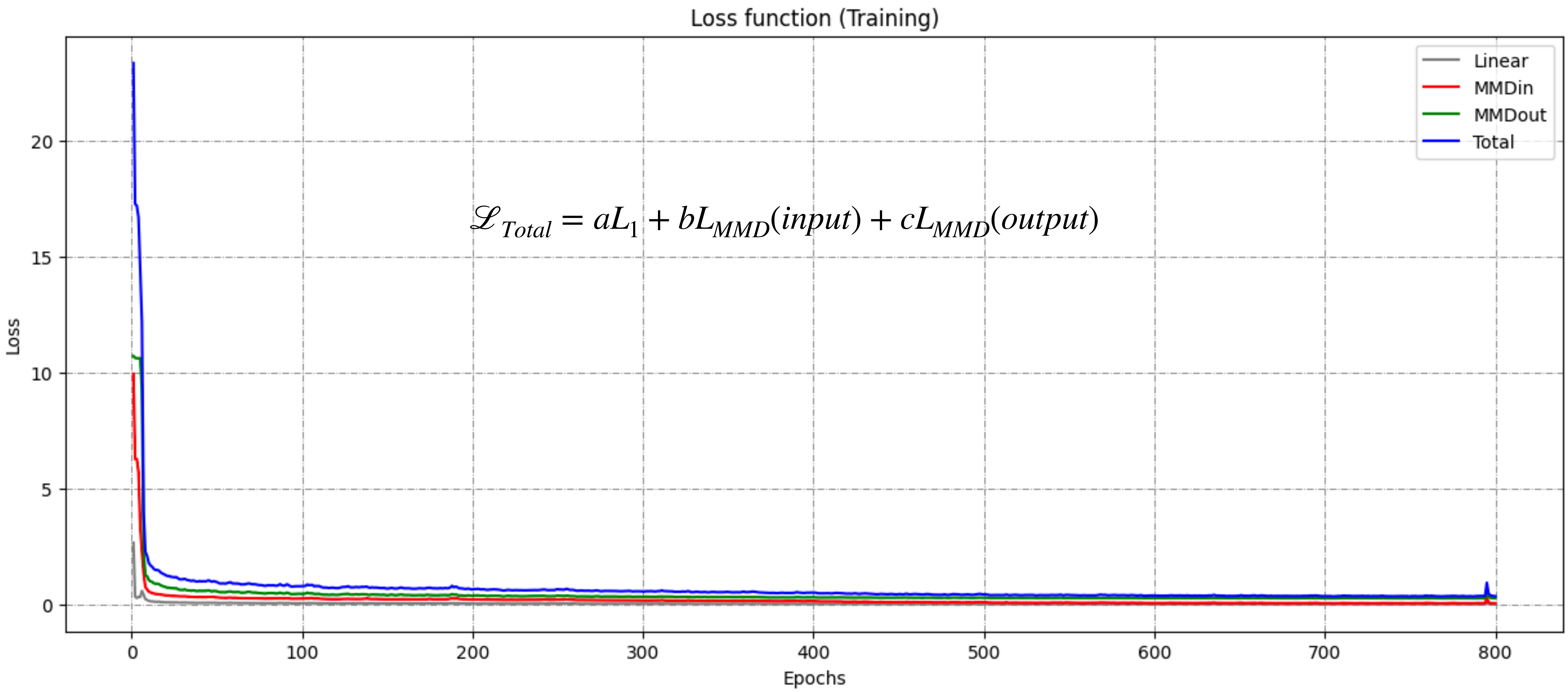
Bjorken variable $x = \dfrac{Q^2}{2P \cdot q}$

Fraction of the momentum

Loss functions

Maximum Mean Discrepancy, and L1

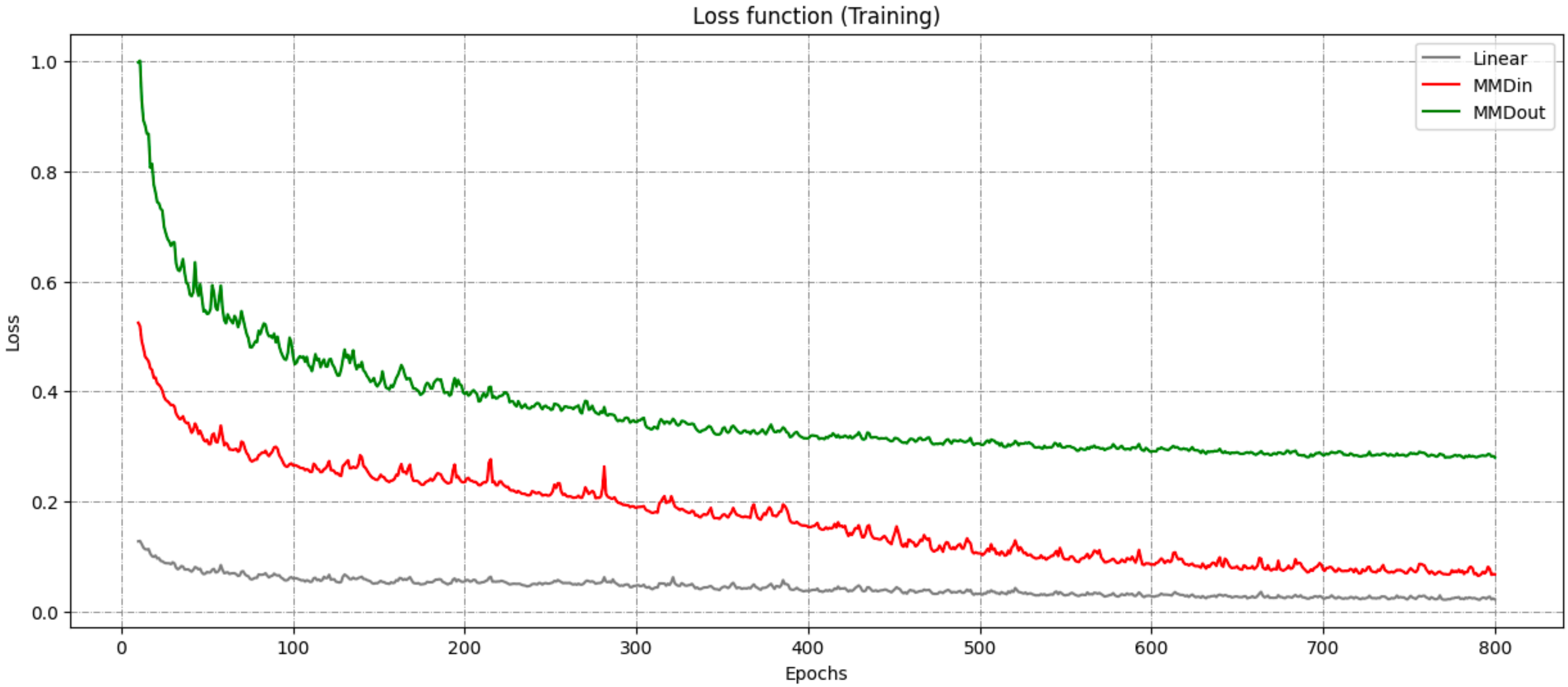
[arXiv:1808.04730]
Analyzing Inverse Problems with Invertible Neural Networks



$$\mathscr{L}_{Total} = aL_1 + bL_{MMD}(input) + cL_{MMD}(output)$$

Loss functions

Maximum Mean Discrepancy, and L1



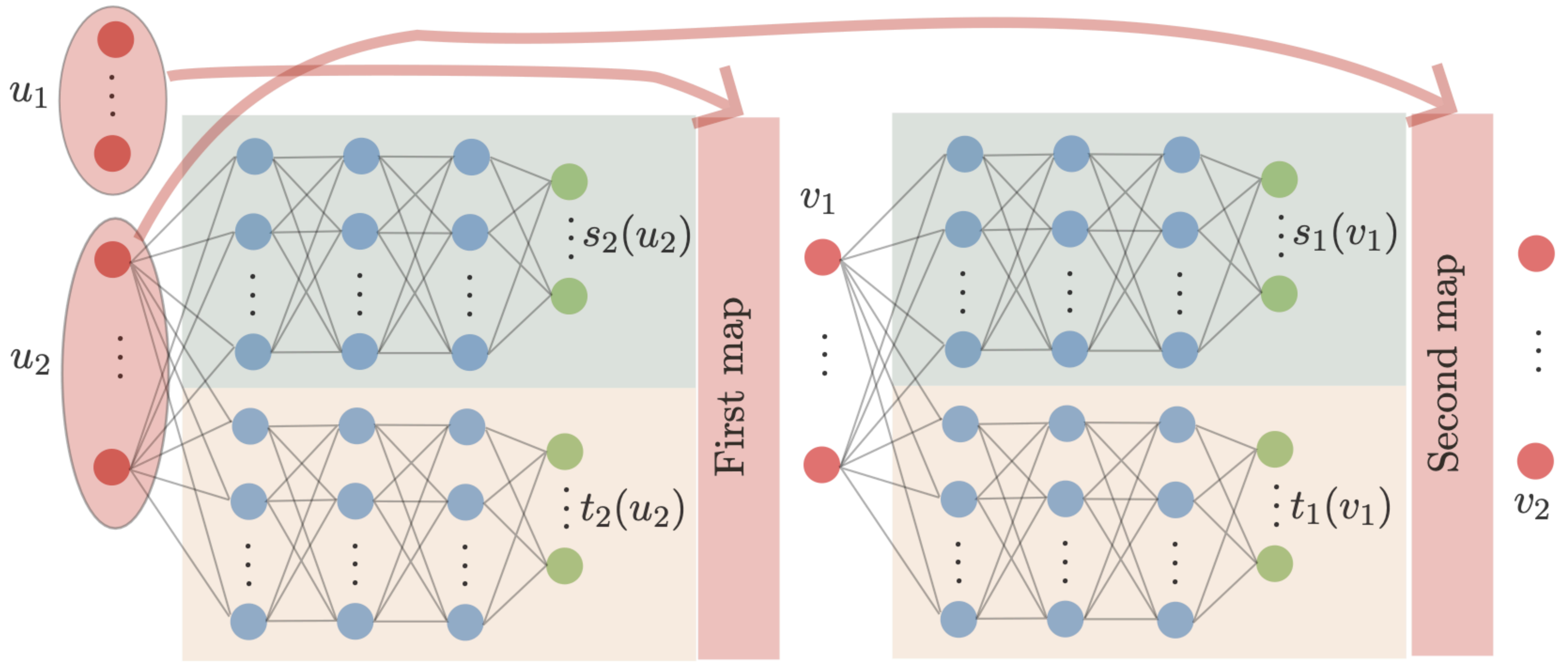
Loss functions

Maximum Mean Discrepancy, and L1

Invertible Neural Networks

Invertible mappings (Affine Coupling Layers)

[arXiv:1605.08803]
Density estimation using Real NVP



$$v_1 = u_1 \odot e^{s_2(u_2)} + t_2(u_2)$$

$$v_2 = u_2 \odot e^{s_1(v_1)} + t_1(v_1)$$

$$x \odot y = (x_1 \cdot y_1, x_2 \cdot y_2)$$

We can calculate the inverse easily!

$$u_1 = e^{-s_2(u_2)} \odot (v_1 - t_2(u_2))$$

$$u_2 = e^{-s_1(v_1)} \odot (v_2 - t_1(v_1))$$