Reconstructing PDFs from LQCD: Invertible Neural Networks

Summer Research Report

Yamil Cahuana Medrano

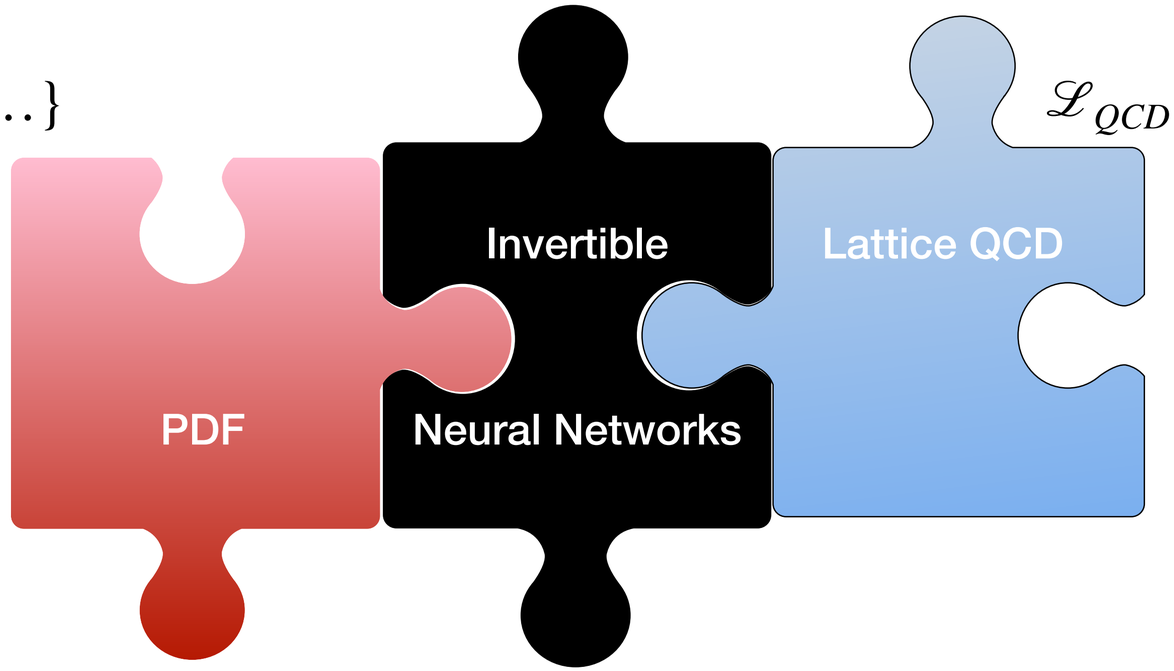
Motivation

PDF ↔ LQCD

- Obtain Parton distribution functions (PDF) from Lattice QCD calculations.
- Solve the inverse problem with INN architecture taking into account data's uncertainty with advanced statistical techniques.

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

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



Parton distribution functions (PDFs)

A brief introduction…

- How can we describe proton's structure?

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

PDF

Unpolarized PDFs — Helicity-averaged

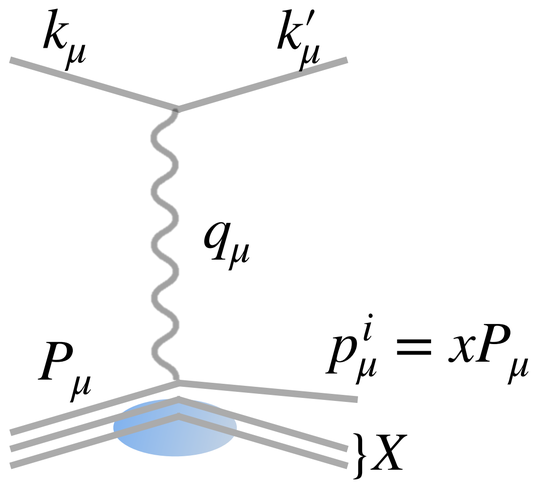
$$f_i(x,\mu^2) = f_i^{\rightarrow}(x,\mu^2) + f_i^{\leftarrow}(x,\mu^2)$$

Polarized PDFs

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

We will focus on this

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



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

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

Fraction of the momentum

PDFs on Euclidean Lattice

Pseudo PDFs, Ioffe-time $\nu = p \cdot z$

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

- Lattice QCD prevents calculations of correlations functions that are in the light-cone coordinates.

Pseudo-ITD

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

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

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

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

$$Q(\nu,\mu^2) = \frac{\mathscr{M}(\nu,z^2)}{\mathscr{M}(0,z^2)} = \int_0^1 \cos(\nu x) q_v(x,\mu^2)$$

Reduced pseudo ITD

Inverse problem if we want to determine $q(x,\mu^2)$.

General description of the problem

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

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

- We have a constraint given by the discretization over nu, which depends on available Lattice data.

$$Q(\nu) = \int_0^1 \cos(\nu x) q_v(x) \approx \int_0^1 \cos(\nu x) f(x) \longrightarrow$$

Discretized version of the Integral

$$\mathcal{Q}_l = \mathcal{C}_{lk} q_k \qquad l = 1,2,\ldots,N_\nu \qquad k = 1,2,\ldots,N_x$$

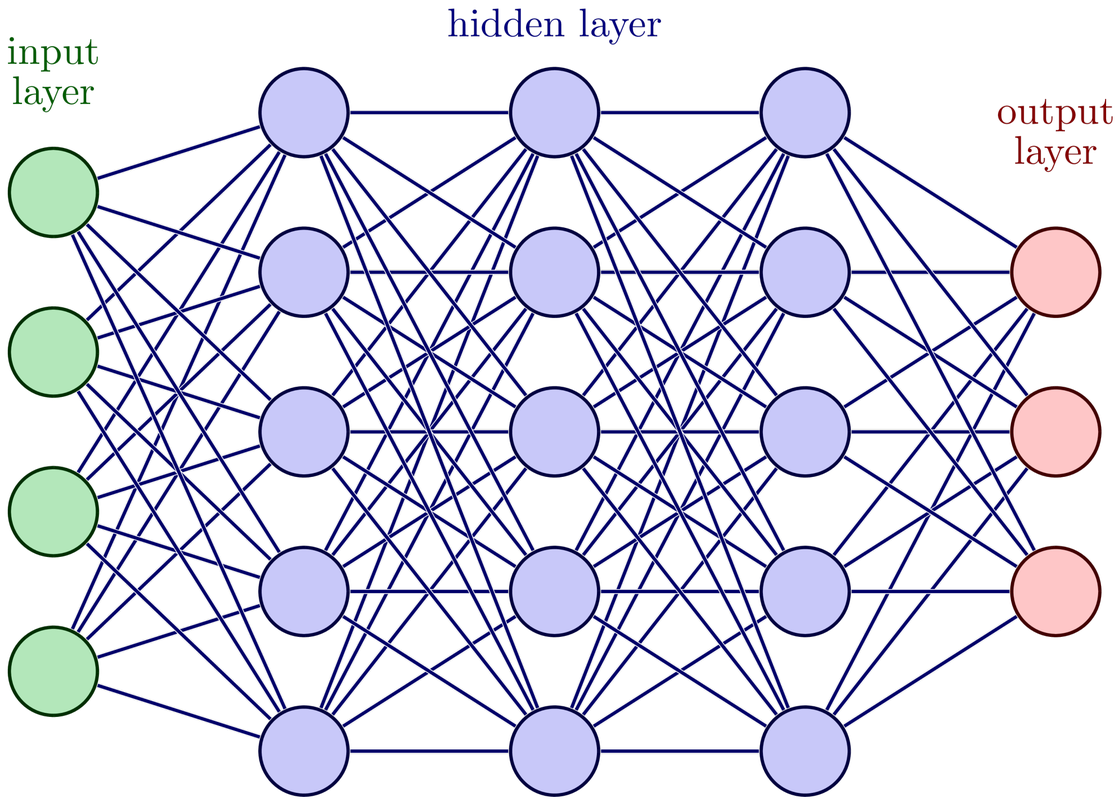
- The matrix $\mathcal{C}_{lk}$ is only invertible if $N_\nu = N_x$, but we want to have a reconstruction of the pdf being independent of N_ν

Interpolation of q

$$f(x) = \frac{x - x_k}{x_{k+1} - x_k} q_v(x_{k+1}) + \frac{x_{k+1} - x}{x_{k+1} - x_k} q_v(x_k) \quad \text{for} \quad x \in [x_k, x_{k+1}] \,.$$

(Invertible) Neural Networks

A short introduction…



- 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

hidden layers

+ Loss Function + Optimizer = INN

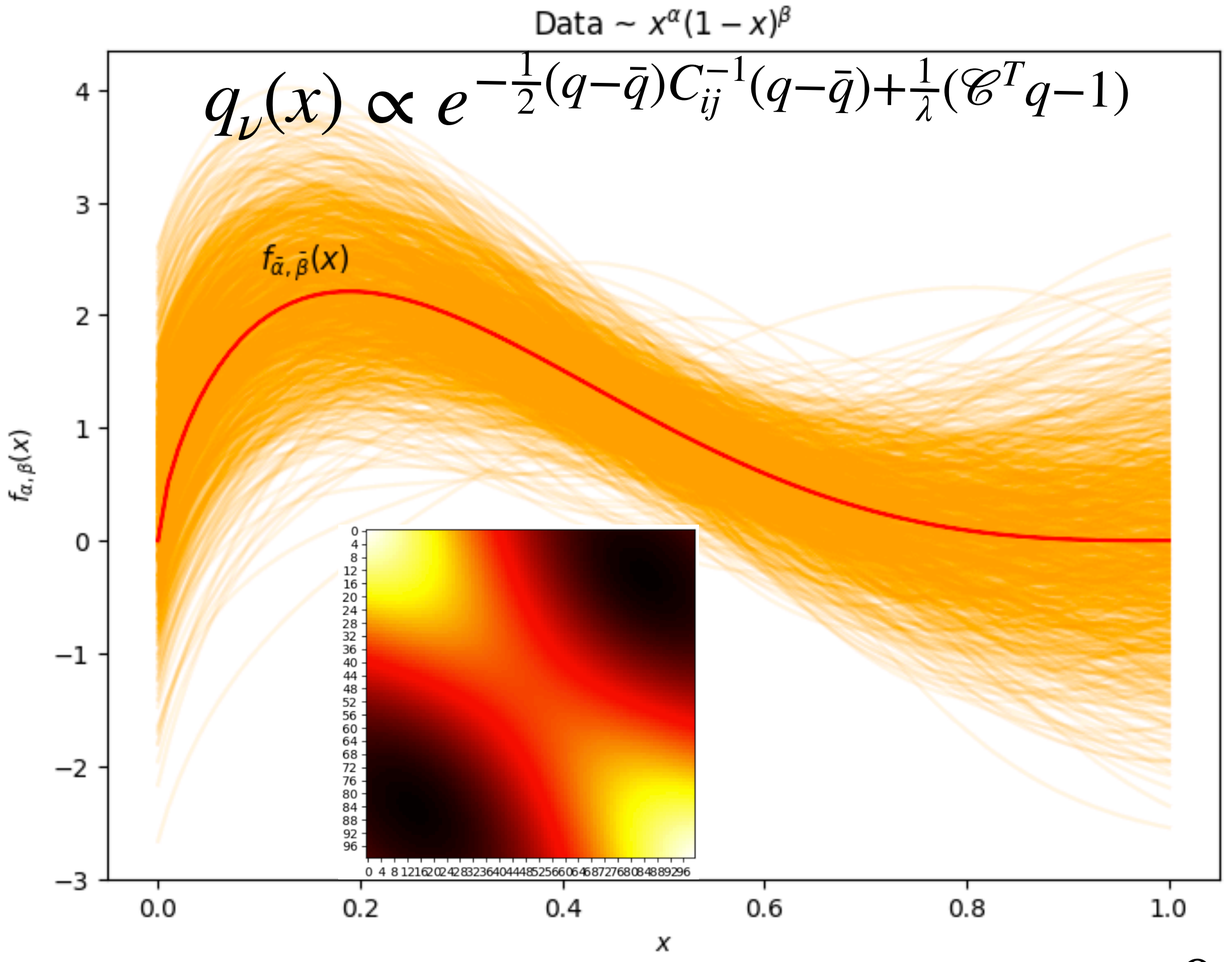
Adam

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

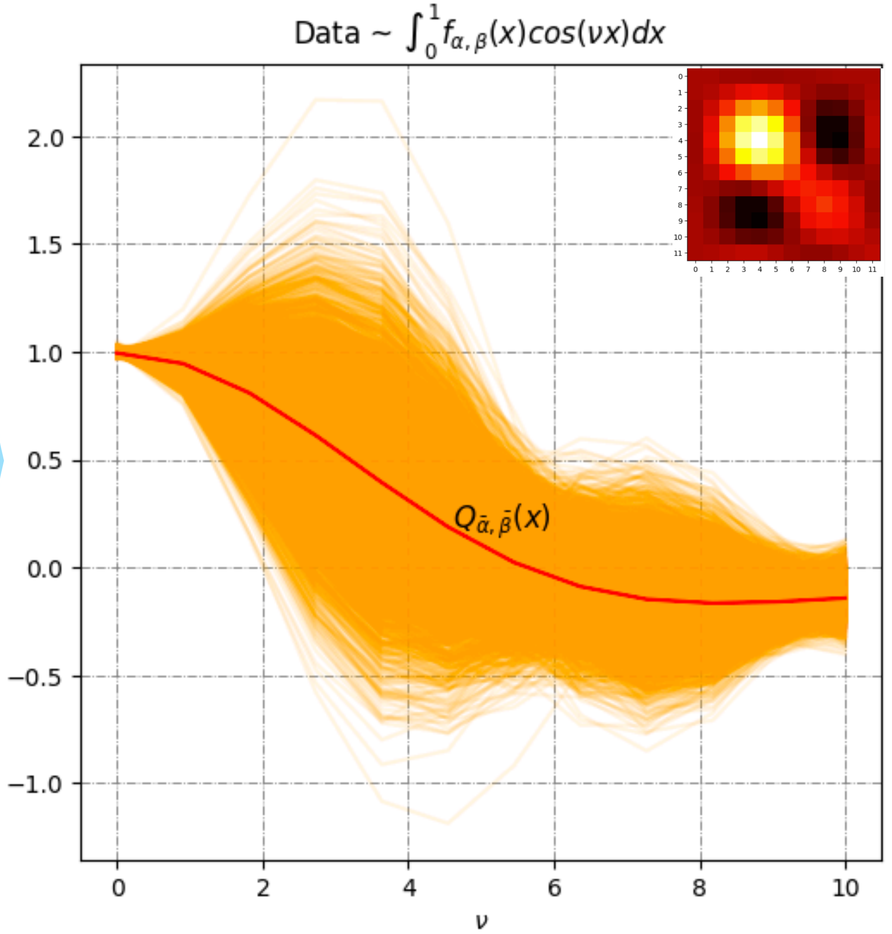
Data

Gaussian process and discretization

- How can we sample $q_\nu(x)$ such that we obtain smooth and normalized curves?



Forward Process (Integration)

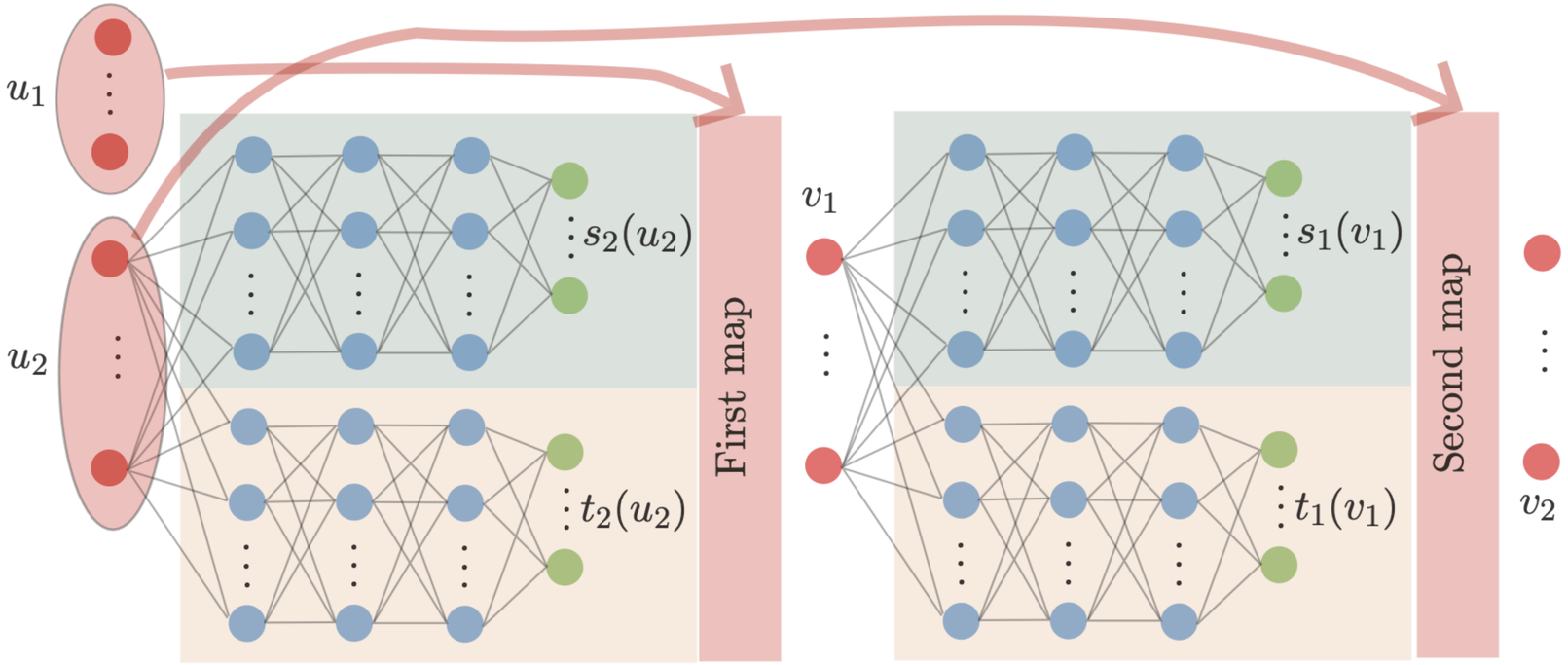


$\alpha = 0.7$ $\beta = 3$

Invertible Neural Networks

Invertible mappings (Affine Coupling Layers)

[arXiv:1605.08803]
Density estimation using Real NVP



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

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

We can calculate the inverse easily!

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

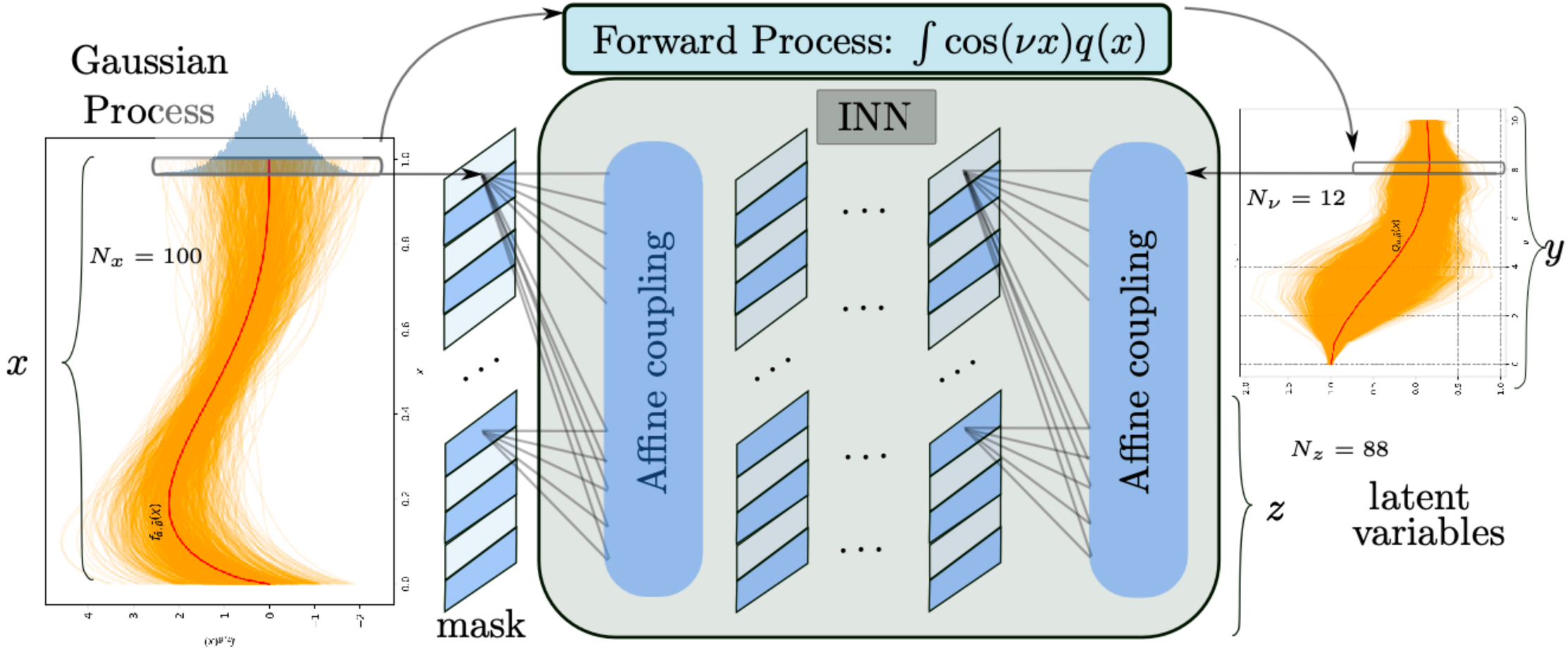
$$u_2 = e^{-s_1(v_1)} \odot (v_2 - t_1(v_1))$$

INNs architecture

Latent variables and discretization

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

- If we stack together blocks of affine couplings, we get INN structure

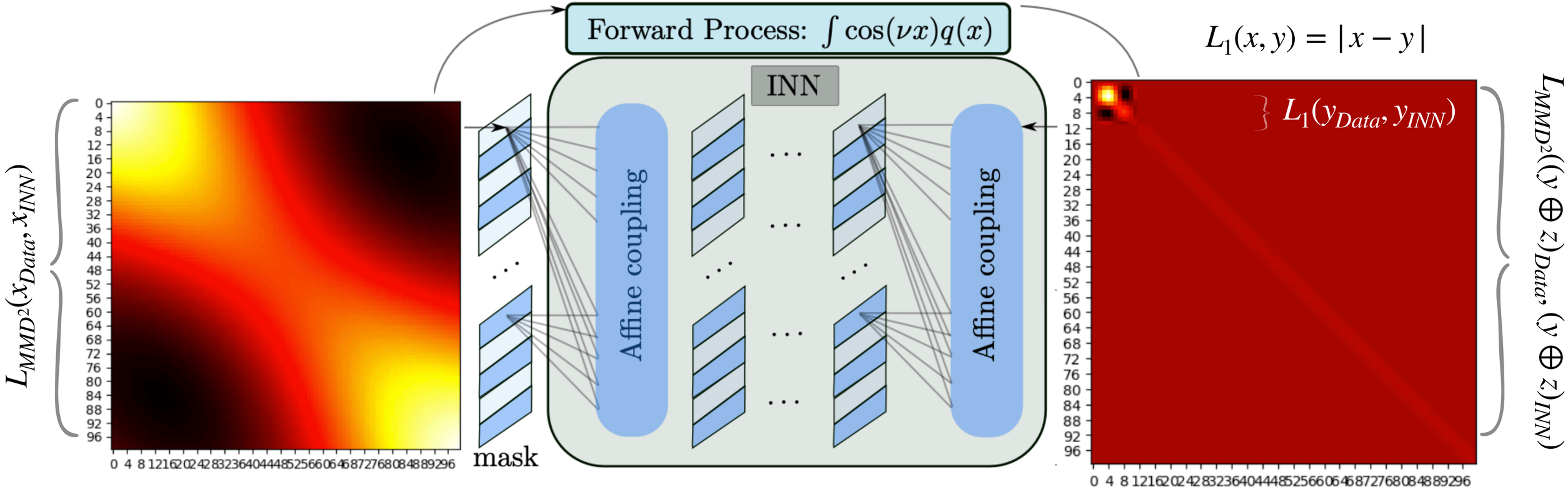


Loss functions

Maximum Mean Discrepancy, and L1

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

- We want to recover probabilities in our neural network mapping.



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

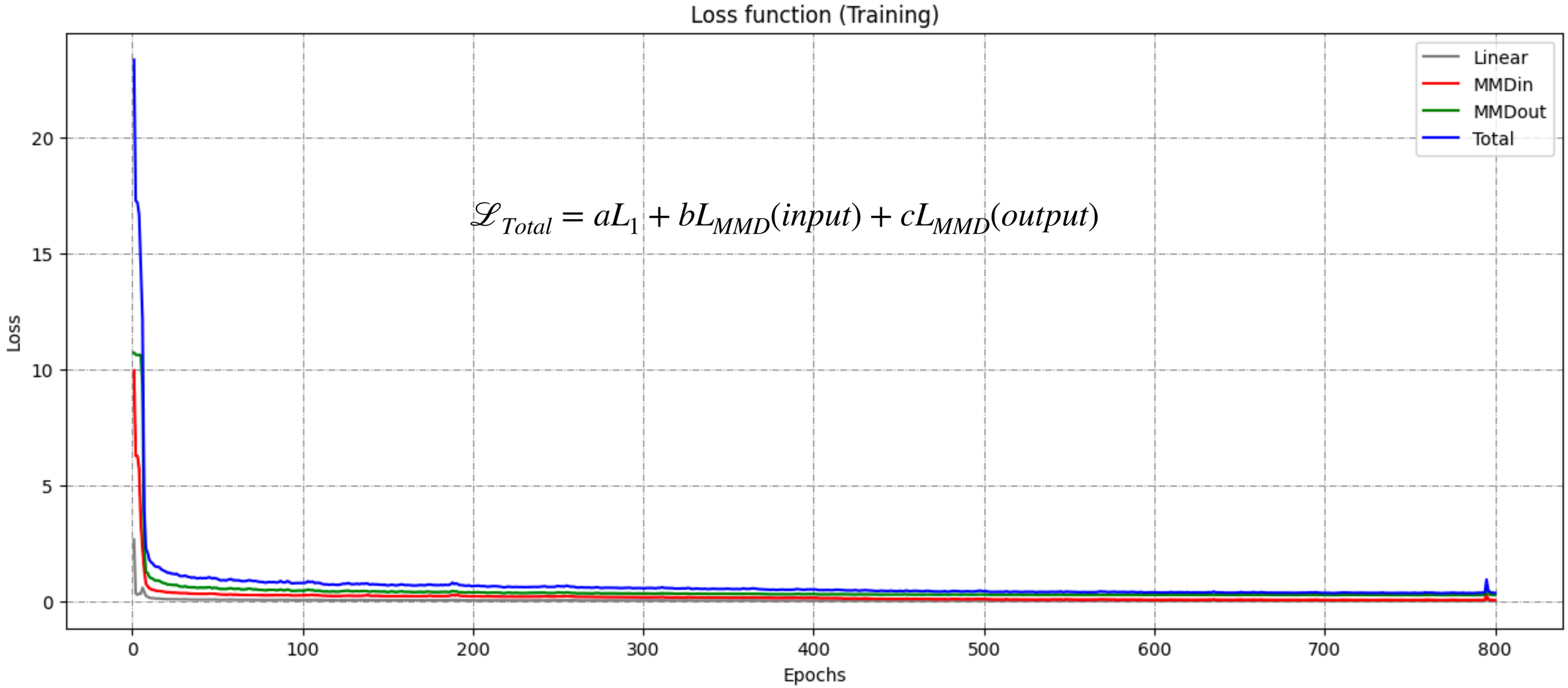
$$L_{MMD^2}(x, y) = \frac{1}{n(n-1)} \sum_{i \neq j} k(x_i, x_j) + \frac{1}{n(n-1)} \sum_{i \neq j} k(y_i, y_j) - \frac{2}{n^2} \sum_{i,j} k(x_i, y_j)$$

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

Loss functions

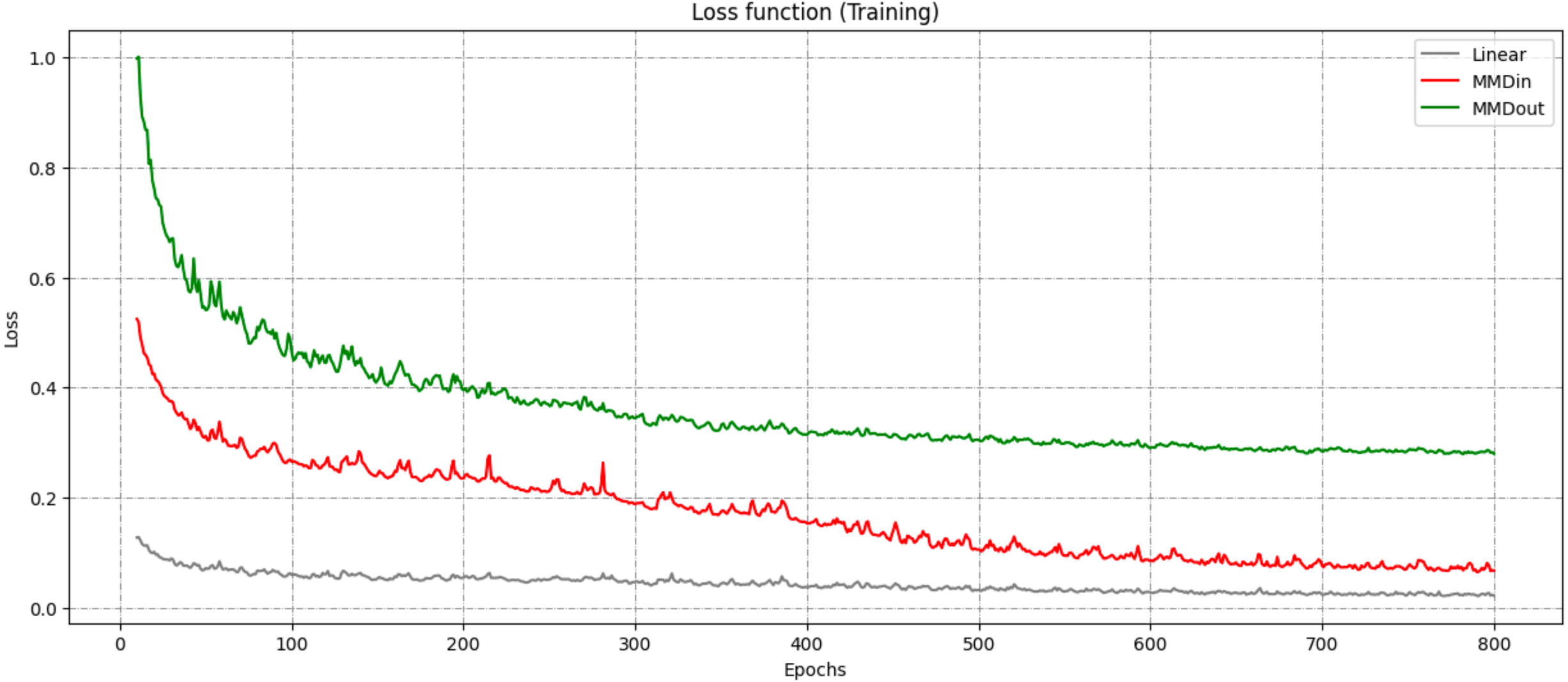
Maximum Mean Discrepancy, and L1

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



Loss functions

Maximum Mean Discrepancy, and L1



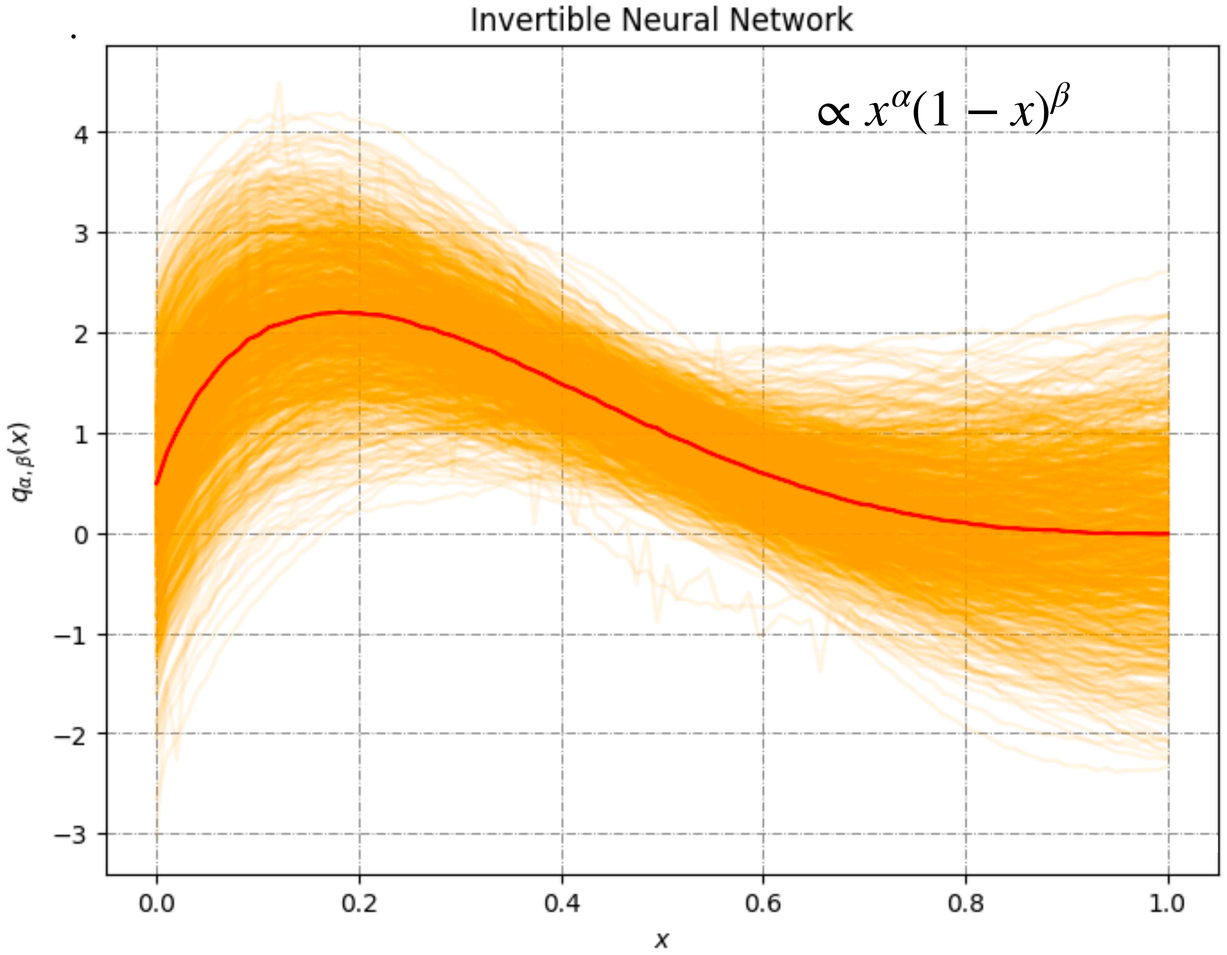
Loss functions

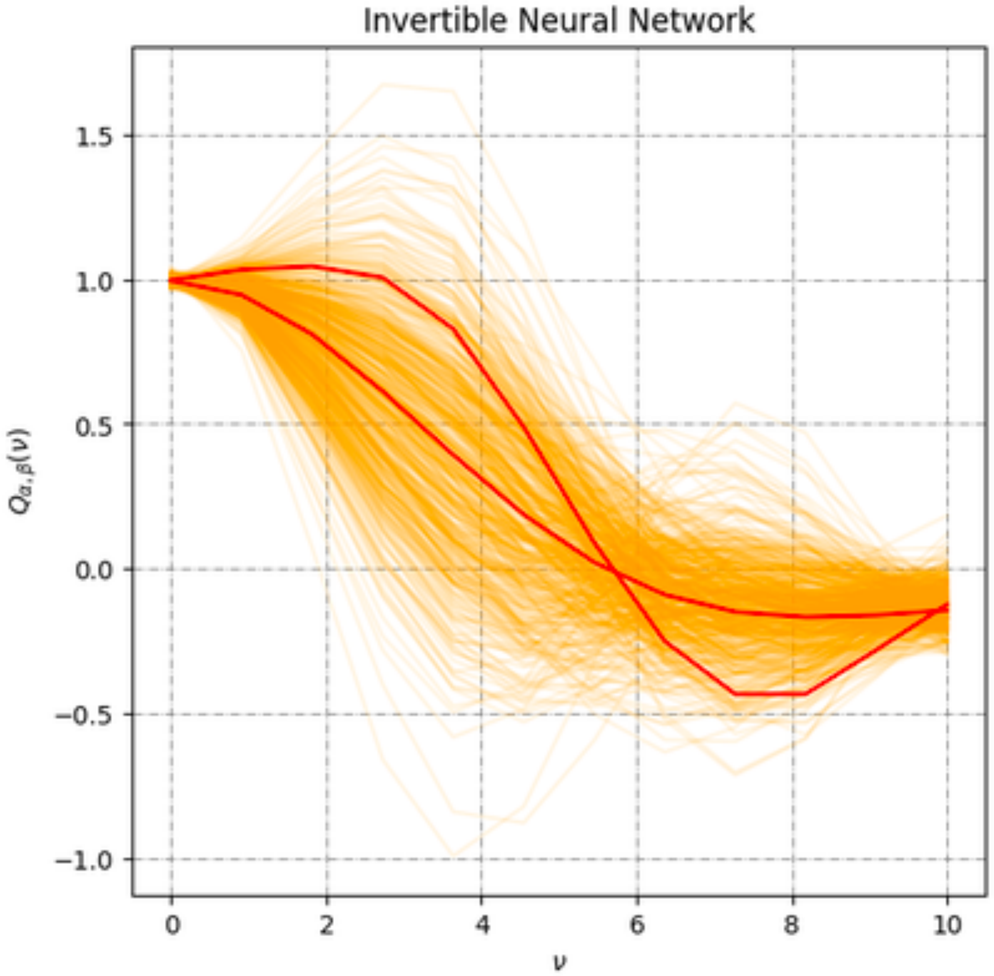
Maximum Mean Discrepancy, and L1

Results

Training data

- Our INN recovers the shape of the input and output

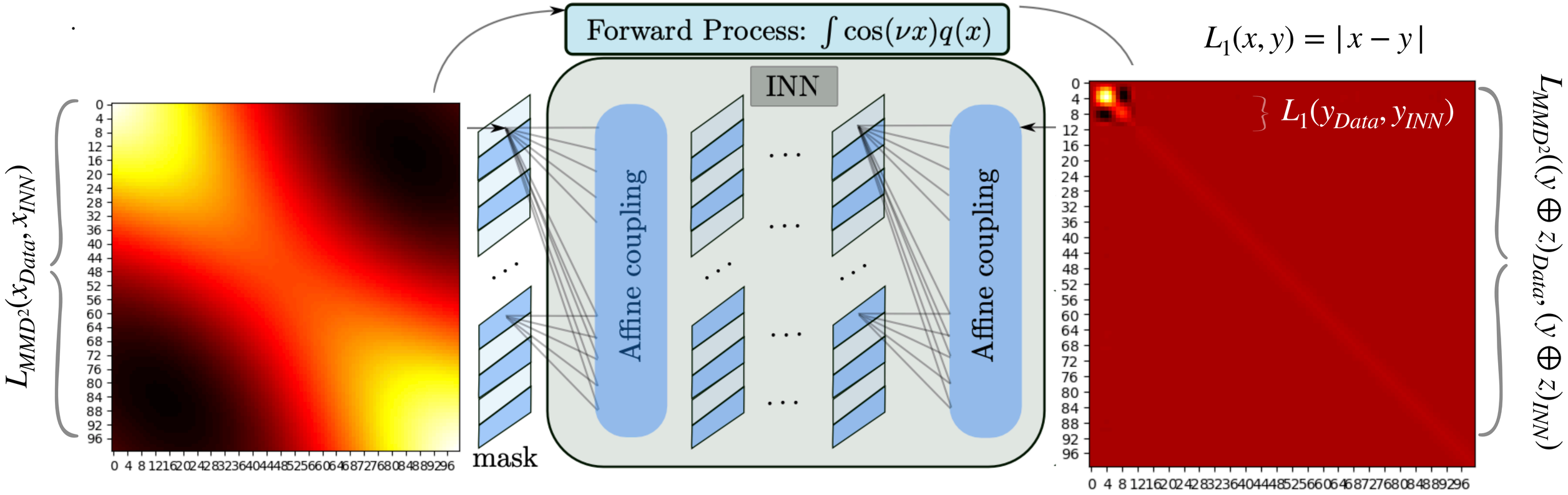




Loss functions (INN generated)

Maximum Mean Discrepancy, and L1

- We did actually recover probability distributions in our input and output.



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

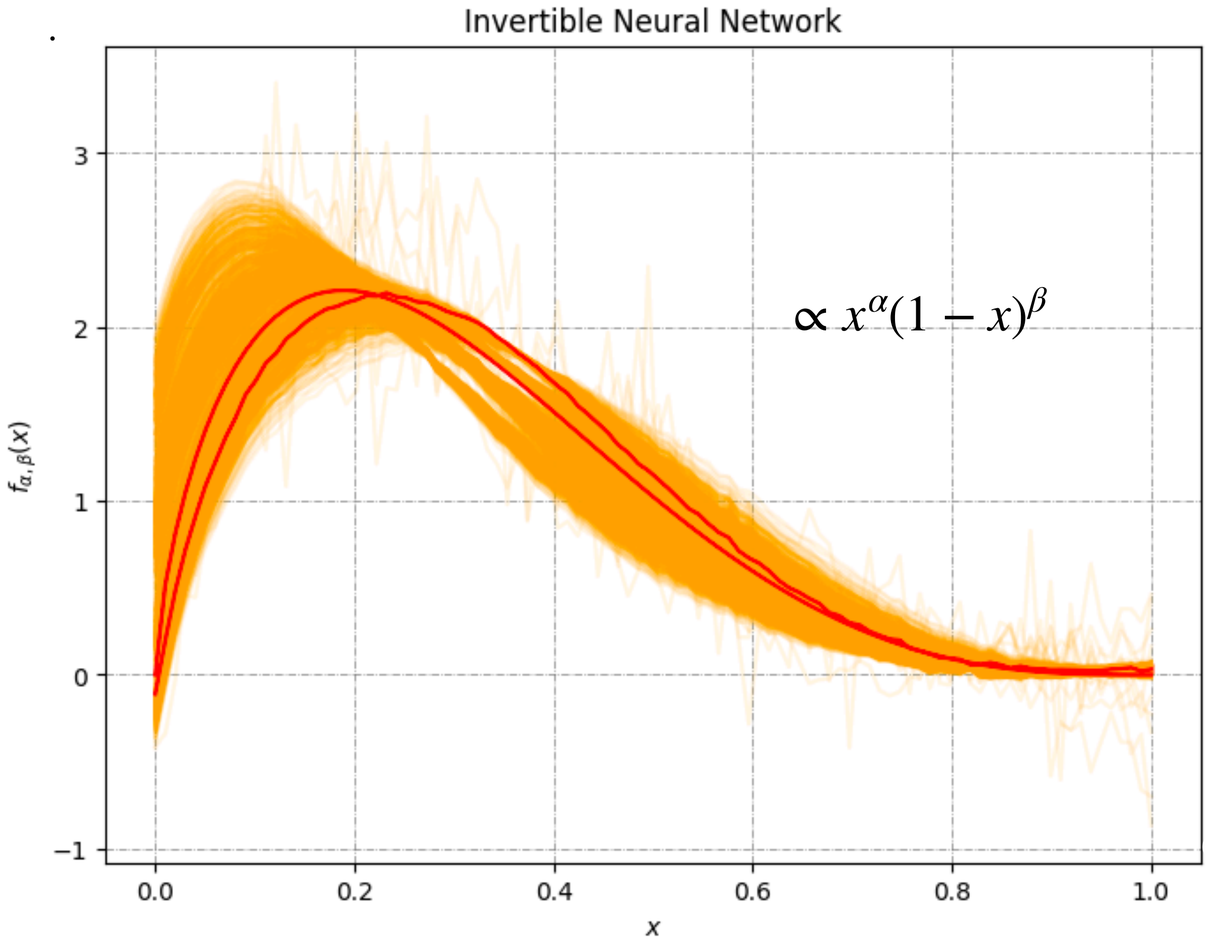
$$k(x, y) = \frac{1}{1 + \frac{|x - y|^2}{h}}$$

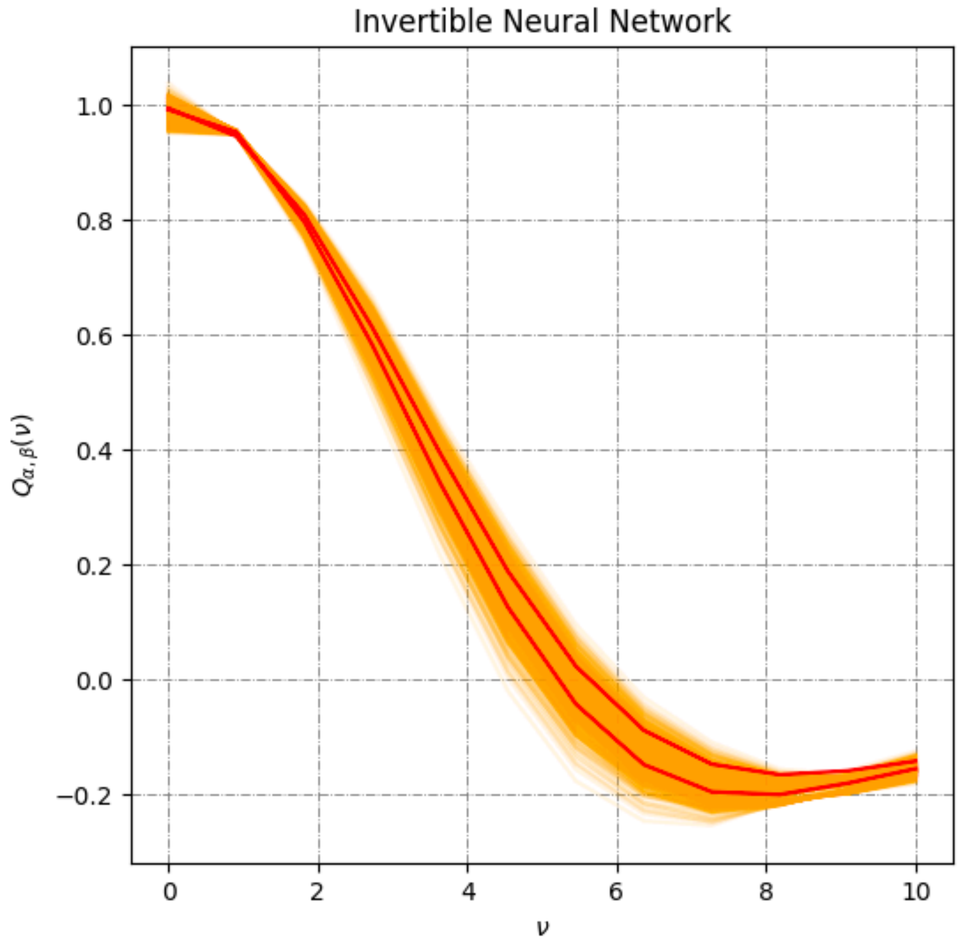
Results

Test data

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

- Data generated by other sampling process.



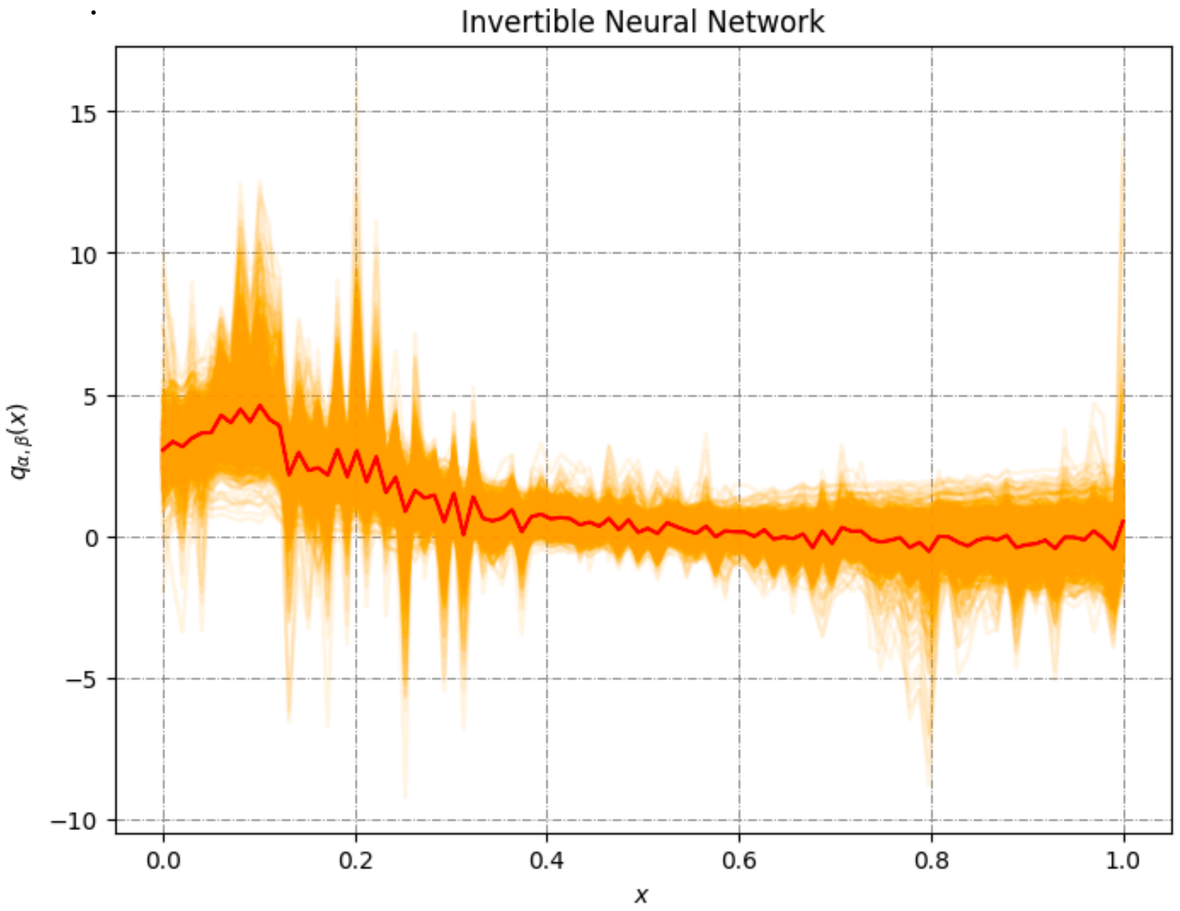


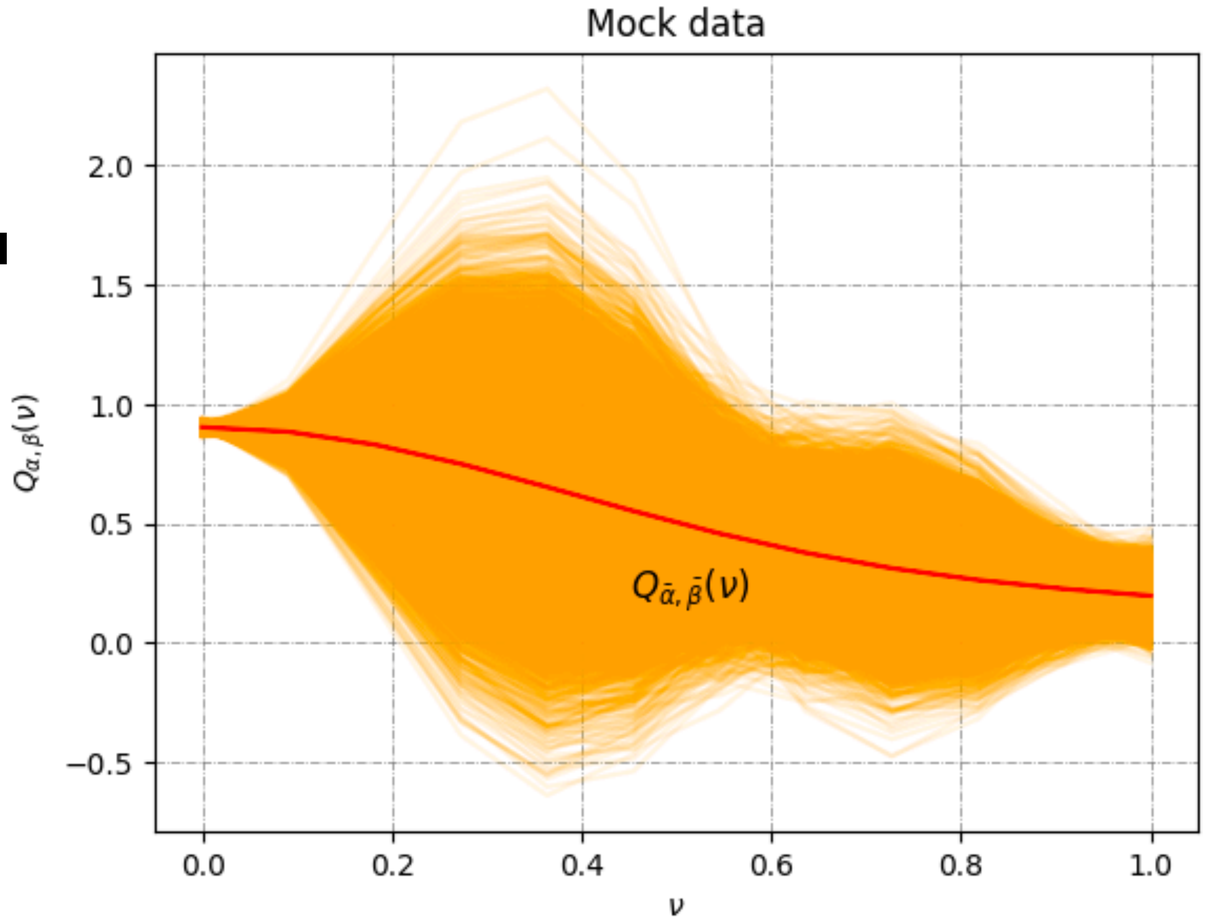
Results

Mock data

- Still working on it…

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



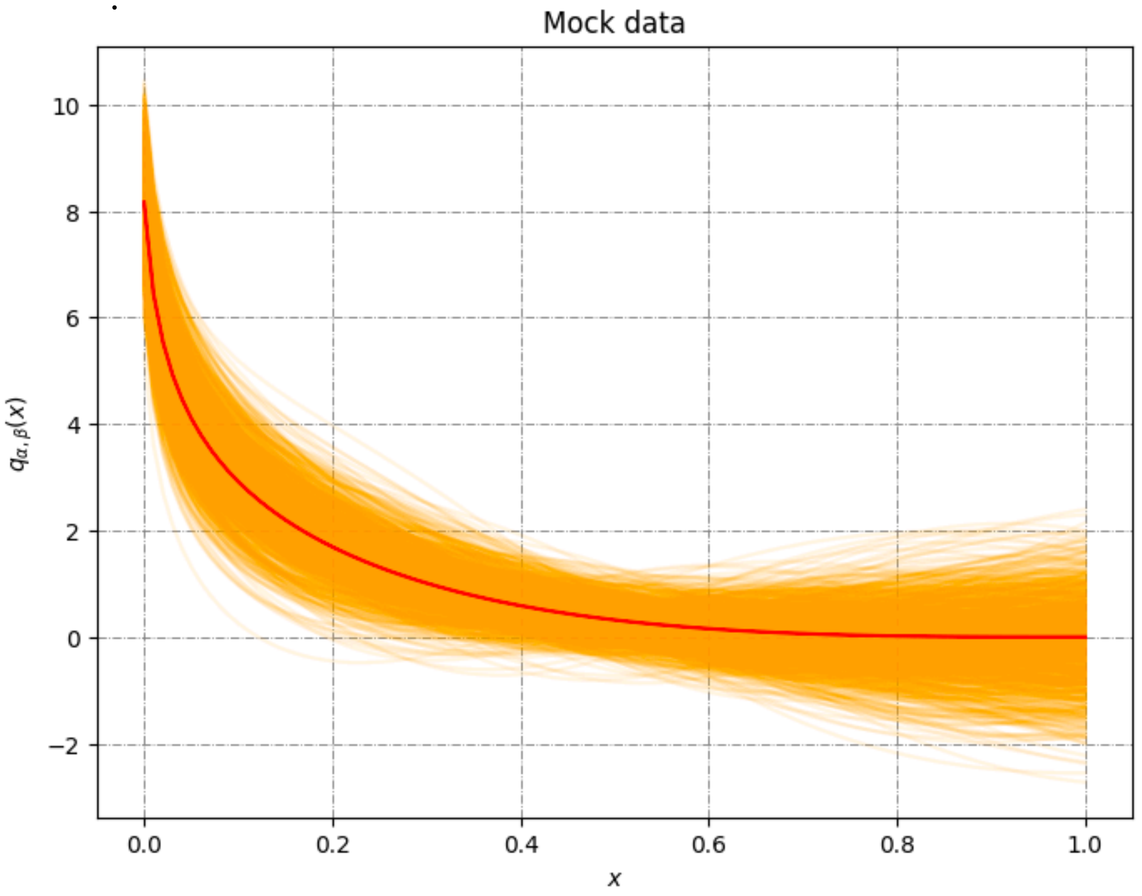


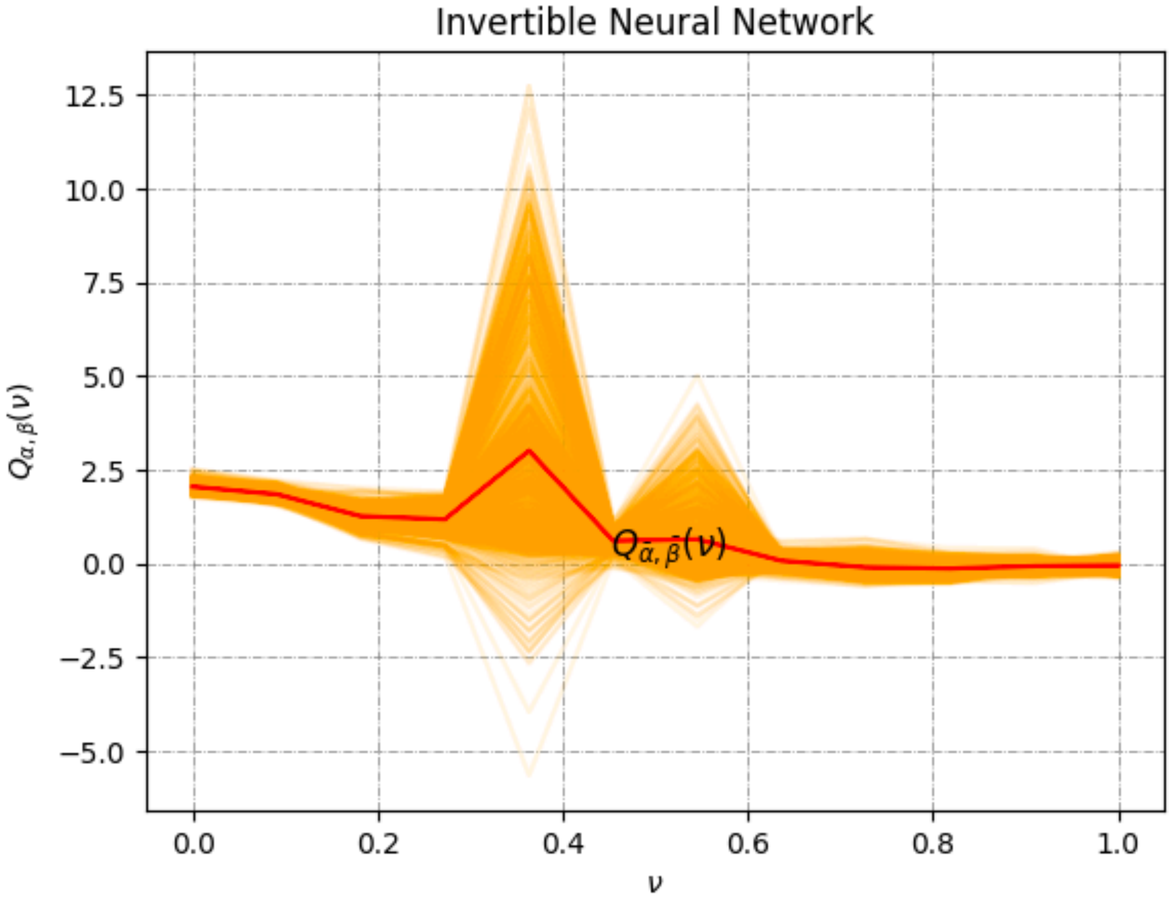
Results

Mock data

- Still working on it…

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





Conclusion

Ways to improve this work…

- Does our INN learns the integration method (Forward Process) and invert it (backward process)?
- We should be able to try different α and β and recover similar results.
- Instead of mock data, use real world data, such as matrix element from Lattice or PDF fits and map one into another.
- Generate hyper-parameters used in gaussian process from Machine Learning optimization methods(GPyTorch).
- Analyze convergence of MMD and check signals of overfitting or under-fitting (basically more epochs).

Future Work…

Normalization Flows in LQCD

- Learn how to sample field configurations

- Use of Normalizing Flows
- The masking process is complex

Quantum simulations of LQCD

- Not yet clear, but working on it.

-Quantum Computing bootcamp: JLab
-2nd Mexican Quantum computing seminar

Trivializing maps

- Basically understand how HMC simulations are benefited from this maps

- Lüscher: Trivializing maps, the Wilson flow and the HMC algorithm

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].