Proximal Mean Field Learning in Shallow Neural Networks

Alexis M.H. Teter

Department of Applied Mathematics
University of California, Santa Cruz

Joint work with

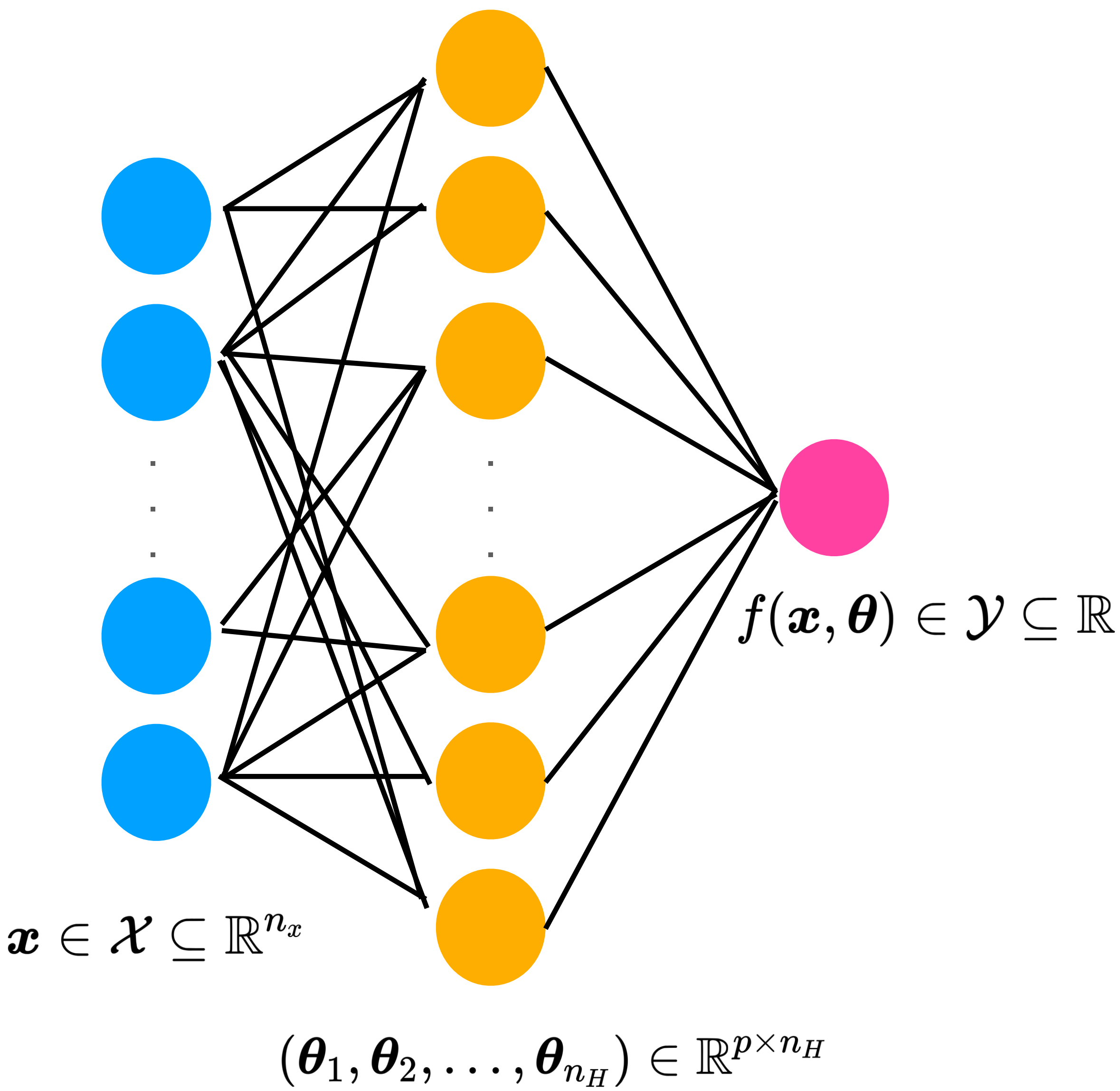
Abhishek Halder
Iowa State University

Iman Nodozi
University of California, Santa Cruz

SIAM Minisymposium: Recent advances in optimization for DNNs

2024 SIAM Annual Meeting, Online, July 18, 2024

Structure of shallow neural network



$$f(\boldsymbol{x}, \boldsymbol{\theta}) := \frac{1}{n_H} \sum_{i=1}^{n_H} \Phi(\boldsymbol{x}, \boldsymbol{\theta}_i)$$

$$\Phi(\boldsymbol{x}, \boldsymbol{\theta}_i) := a_i \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle + b_i)$$

Risk

Population Risk

$$R(f) := \mathbb{E}_{(\boldsymbol{x},y)\sim\gamma}[(y - f(\boldsymbol{x}, \bar{\boldsymbol{\theta}}))^2]$$

Empirical Risk

$$R(f) \approx \frac{1}{n_{\mathrm{data}}} \sum_{j=1}^{n_{\mathrm{data}}} (y_j - f(\boldsymbol{x}_j, \bar{\boldsymbol{\theta}}))^2$$

Large dimensional, non convex optimization problem

$$\min_{\bar{\boldsymbol{\theta}} \in \mathbb{R}^{p \times n_{\mathrm{H}}}} R(f)$$

Mean field limit

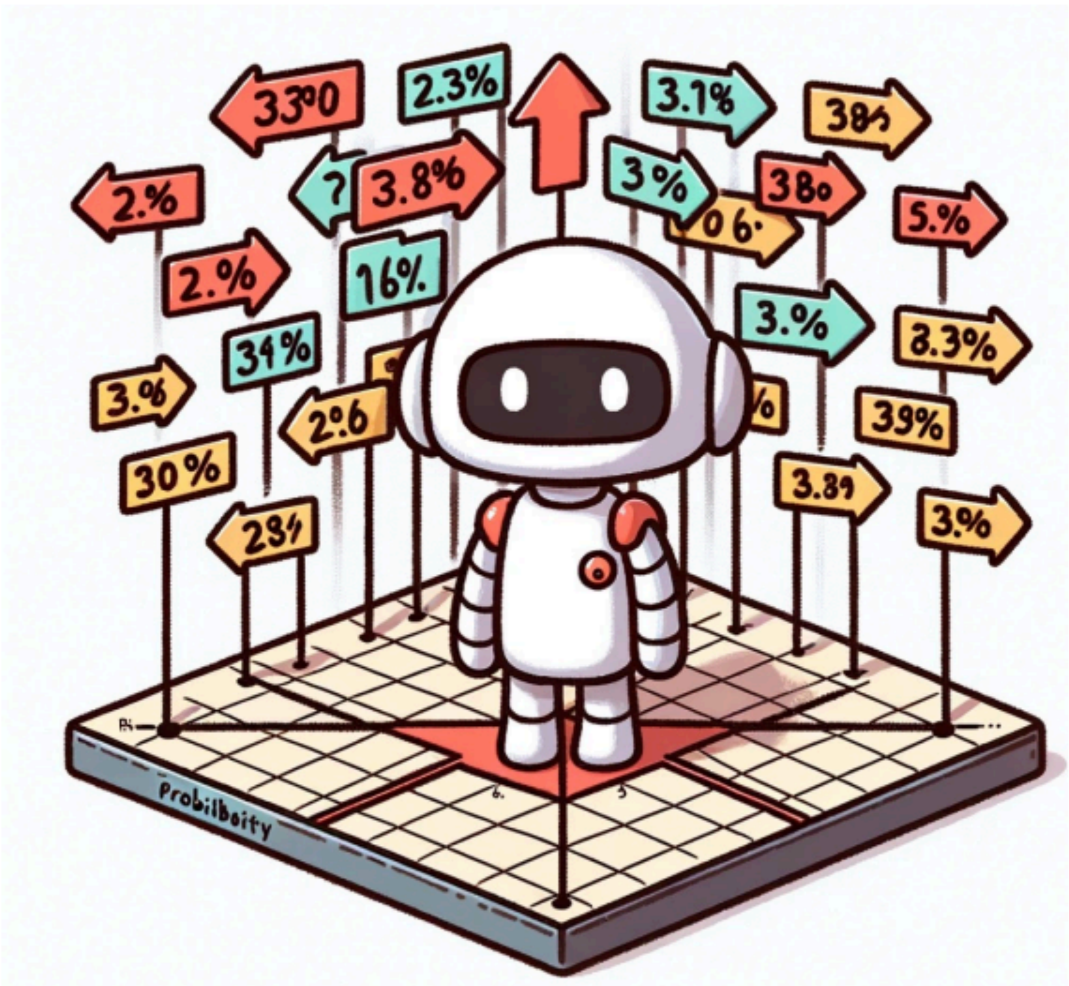
$$f(\boldsymbol{x}, \boldsymbol{\theta}) := \frac{1}{n_H} \sum_{i=1}^{n_H} \Phi(\boldsymbol{x}, \boldsymbol{\theta}_i)$$

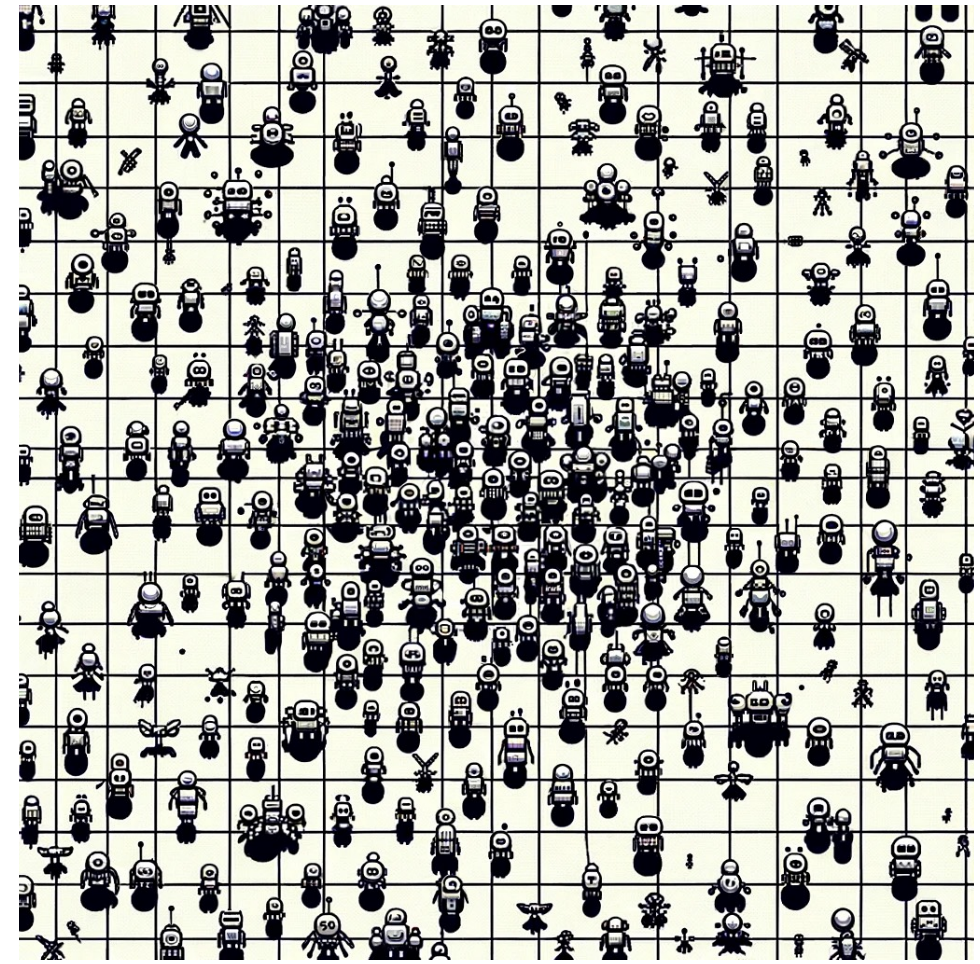
Let $n_H \to \infty$

$$f_{\text{MeanField}} := \int_{\mathbb{R}^p} \Phi(\boldsymbol{x}, \boldsymbol{\theta}) d\mu(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}}[\Phi(\boldsymbol{x}, \boldsymbol{\theta})]$$

where

$$d\mu(\boldsymbol{\theta}) = \rho(\boldsymbol{\theta}) d\boldsymbol{\theta}$$

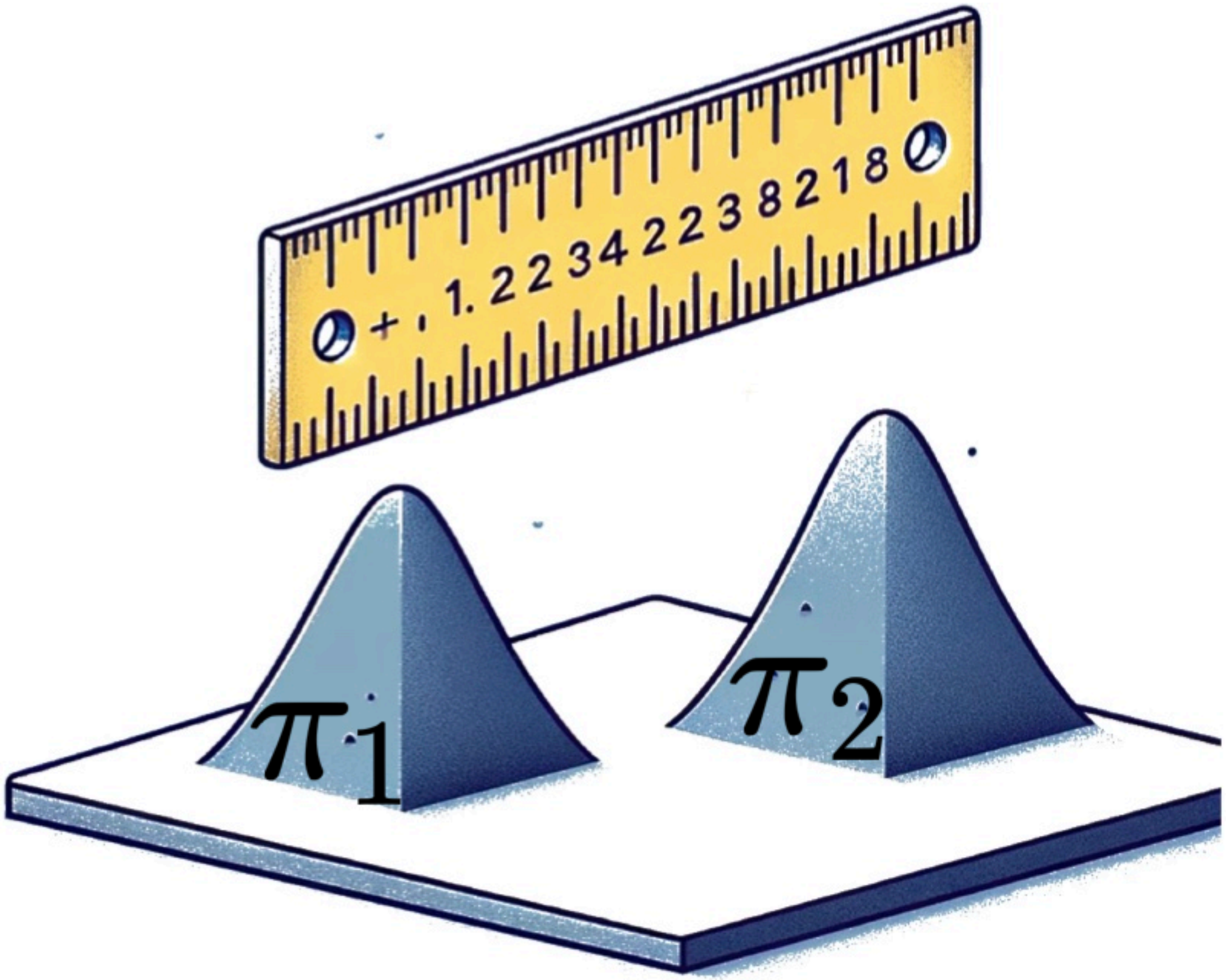




$$\boxed{d\mu = \rho \, dx}$$

Images generated with ChatGPT by Iman Nodozi

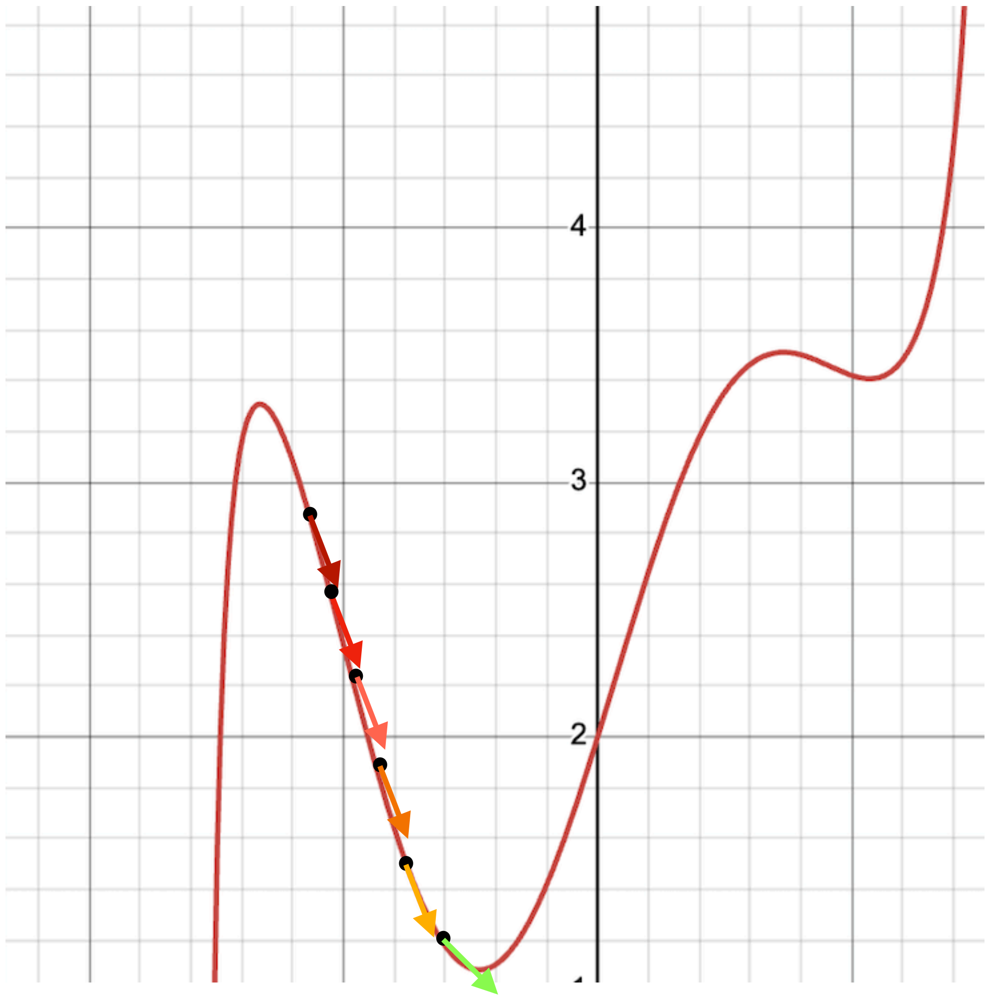
Wasserstein metric



$$W_2^2(\pi_1, \pi_2) := \inf_{\pi \in \Pi(\pi_1, \pi_2)} \int_{\mathcal{Z}_1 \times \mathcal{Z}_2} \|\boldsymbol{z}_1 - \boldsymbol{z}_2\|_2^2 \, d\pi(\boldsymbol{z}_1, \boldsymbol{z}_2)$$

Image generated with ChatGPT by Iman Nodozi

Gradient Flows



Gradient Flow

$$\frac{d\boldsymbol{x}}{dt} = -\nabla f(\boldsymbol{x}), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0$$

Recursion

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} - h\nabla f(\boldsymbol{x}_k) = \underset{\boldsymbol{x}\in\mathcal{R}^n}{\arg\min}\left(\frac{1}{2}|\boldsymbol{x} - \boldsymbol{x}_{k-1}|_2^2 + hf(\boldsymbol{x})\right) = \mathrm{prox}_{hF}^{|\cdot|_2}(\boldsymbol{x}_{k-1})$$

proximal update

$$= \underset{\text{decision variable}}{\arg\inf}\left\{\frac{1}{2}\mathrm{dist}^2(\text{decision variable, input}) + \text{time step} \times \text{functional}(\text{decision variable})\right\}$$

Gradient Flows

Gradient Flow

$$\frac{\partial \rho}{\partial t} = -\nabla^{W_2} F(\rho) := -\nabla \cdot \left(\rho \nabla \frac{\delta F}{\partial \rho}\right) \quad \textbf{where} \quad \rho(\boldsymbol{\theta}, 0) = \rho_0(\boldsymbol{\theta})$$

Recursion

$$\rho_k = \rho(\cdot, t = kh) = \underset{\rho \in \mathcal{P}_2(\mathbb{R}^p)}{\arg\min} \left(\frac{1}{2} W_2^2(\rho, \rho_{k-1}) + hF(\rho) \right) = \text{prox}_{hF}^{W_2}(\rho_{k-1})$$

proximal update

$$= \underset{\text{decision variable}}{\arg\inf} \left\{ \frac{1}{2} \text{dist}^2(\text{decision variable, input}) + \text{time step} \times \text{functional(decision variable)} \right\}$$

Risk functional

$$R(f_{\text{Mean Field}}(\boldsymbol{x},\rho)) = \mathbb{E}_{(\boldsymbol{x},y)}\left(y - \int_{\mathbb{R}^p} \Phi(\boldsymbol{x},\theta)\rho(\boldsymbol{\theta})d\boldsymbol{\theta}\right)^2$$

$$R(f_{\text{Mean Field}}(\boldsymbol{x},\rho)) = F_0 + \int_{\mathbb{R}^p} \underbrace{V(\boldsymbol{\theta})}_{\text{Drift potential}}\rho(\boldsymbol{\theta})d\boldsymbol{\theta} + \int_{\mathbb{R}^{2p}} \underbrace{U(\boldsymbol{\theta},\tilde{\boldsymbol{\theta}})}_{\text{Interaction potential}}\rho(\boldsymbol{\theta})\rho(\tilde{\boldsymbol{\theta}})d\boldsymbol{\theta}d\tilde{\boldsymbol{\theta}}$$

$$F_0 := \mathbb{E}_{(\boldsymbol{x},y)}[y^2]$$

$$V(\boldsymbol{\theta}) := \mathbb{E}_{(\boldsymbol{x},y)}[-2y\,\Phi(\boldsymbol{x},\boldsymbol{\theta})]$$

$$U(\boldsymbol{\theta},\tilde{\boldsymbol{\theta}}) := \mathbb{E}_{(\boldsymbol{x},y)}[\Phi(\boldsymbol{x},\boldsymbol{\theta})\Phi(\boldsymbol{x},\tilde{\boldsymbol{\theta}})]$$

Supervised learning in mean field limit

Supervised learning problem

$$\min_{\rho} \; F(\rho) := \min_{\rho} \; R(f_{\text{Mean Field}}(\boldsymbol{x}, \rho))$$

$$R(f_{\text{Mean Field}}(\boldsymbol{x}, \rho)) = \underbrace{F_0}_{\text{independent of } \rho} + \underbrace{\int_{\mathbb{R}^p} V(\boldsymbol{\theta})\rho(\boldsymbol{\theta})d\boldsymbol{\theta}}_{\text{potential energy; linear in } \rho} + \underbrace{\int_{\mathbb{R}^{2p}} U(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}})\rho(\boldsymbol{\theta})\rho(\tilde{\boldsymbol{\theta}})d\boldsymbol{\theta}d\tilde{\boldsymbol{\theta}}}_{\text{bilinear interaction energy; nonlinear in } \rho}$$

With convex regularizer

Mei et al. (2018)

Regularized risk functional

$$F_\beta(\rho) := F(\rho) + \beta^{-1} \int_{\mathbb{R}^p} \rho \log \rho \, d\boldsymbol{\theta}, \qquad \beta > 0$$

$$\rho^{\mathrm{opt}}(t, \boldsymbol{\theta})$$

Nonlinear PDE IVP

$$\frac{\partial \rho}{\partial t} = \nabla_{\boldsymbol{\theta}} \cdot \left(\rho \left(V(\boldsymbol{\theta}) + \int_{\mathbb{R}^p} U(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) \rho(\tilde{\boldsymbol{\theta}}) d\tilde{\boldsymbol{\theta}} \right) \right) + \beta^{-1} \Delta_{\boldsymbol{\theta}} \rho$$

where $\rho(t = 0, \boldsymbol{\theta}) = \rho_0$

Proximal recursions

Proximal recursion

$$\varrho_k = \mathrm{prox}^{W_2}_{hF_\beta}(\varrho_{k-1}) := \underset{\varrho \in \mathcal{P}_2(\mathbb{R}^p)}{\mathrm{arginf}} \left\{ \frac{1}{2} \left(W_2(\varrho, \varrho_{k-1})\right)^2 + h\, F_\beta(\varrho) \right\}$$

$$\textbf{where}\quad \varrho_{k-1}(\cdot) := \varrho(\cdot, t_{k-1})$$

$$\varrho_0 \equiv \rho_0$$

Approximate bilinear term as...

$$\int_{\mathbb{R}^{2p}} U(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) \varrho(\boldsymbol{\theta}) \varrho(\tilde{\boldsymbol{\theta}}) d\boldsymbol{\theta} d\tilde{\boldsymbol{\theta}} \approx \int_{\mathbb{R}^{2p}} U(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) \varrho(\boldsymbol{\theta}) \varrho_{k-1}(\tilde{\boldsymbol{\theta}}) d\boldsymbol{\theta} d\tilde{\boldsymbol{\theta}}$$

(Benamou et al., 2016, Sec. 4)

Proximal recursions

Proximal recursion

$$\varrho_k = \text{prox}_{hF_\beta}^{W_2}(\varrho_{k-1}) := \underset{\varrho \in \mathcal{P}_2(\mathbb{R}^p)}{\text{arginf}} \left\{ \frac{1}{2} (W_2(\varrho, \varrho_{k-1}))^2 + h\, \hat{F}_\beta(\varrho) \right\}$$

where $\varrho_{k-1}(\cdot) := \varrho(\cdot, t_{k-1})$

$$\varrho_0 \equiv \rho_0$$

Approximation of regularized risk functional

$$\hat{F}_\beta(\varrho, \varrho_{k-1}) := \int_{\mathbb{R}^p} \left(F_0 + V(\boldsymbol{\theta}) + \left(\int_{\mathbb{R}^p} U(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) \varrho_{k-1}(\tilde{\boldsymbol{\theta}}) d\tilde{\boldsymbol{\theta}} \right) + \beta^{-1} \log \varrho(\boldsymbol{\theta}) \right) \varrho(\boldsymbol{\theta}) d\boldsymbol{\theta}$$

Proximal recursions

Proximal recursion (semi-implicit variant)

$$\varrho_k = \mathrm{prox}_{hF_\beta}^{W_2}(\varrho_{k-1}) := \underset{\varrho \in \mathcal{P}_2(\mathbb{R}^p)}{\arg\inf} \left\{ \frac{1}{2}\left(W_2(\varrho, \varrho_{k-1})\right)^2 + h\, \hat{F}_\beta(\varrho) \right\}$$

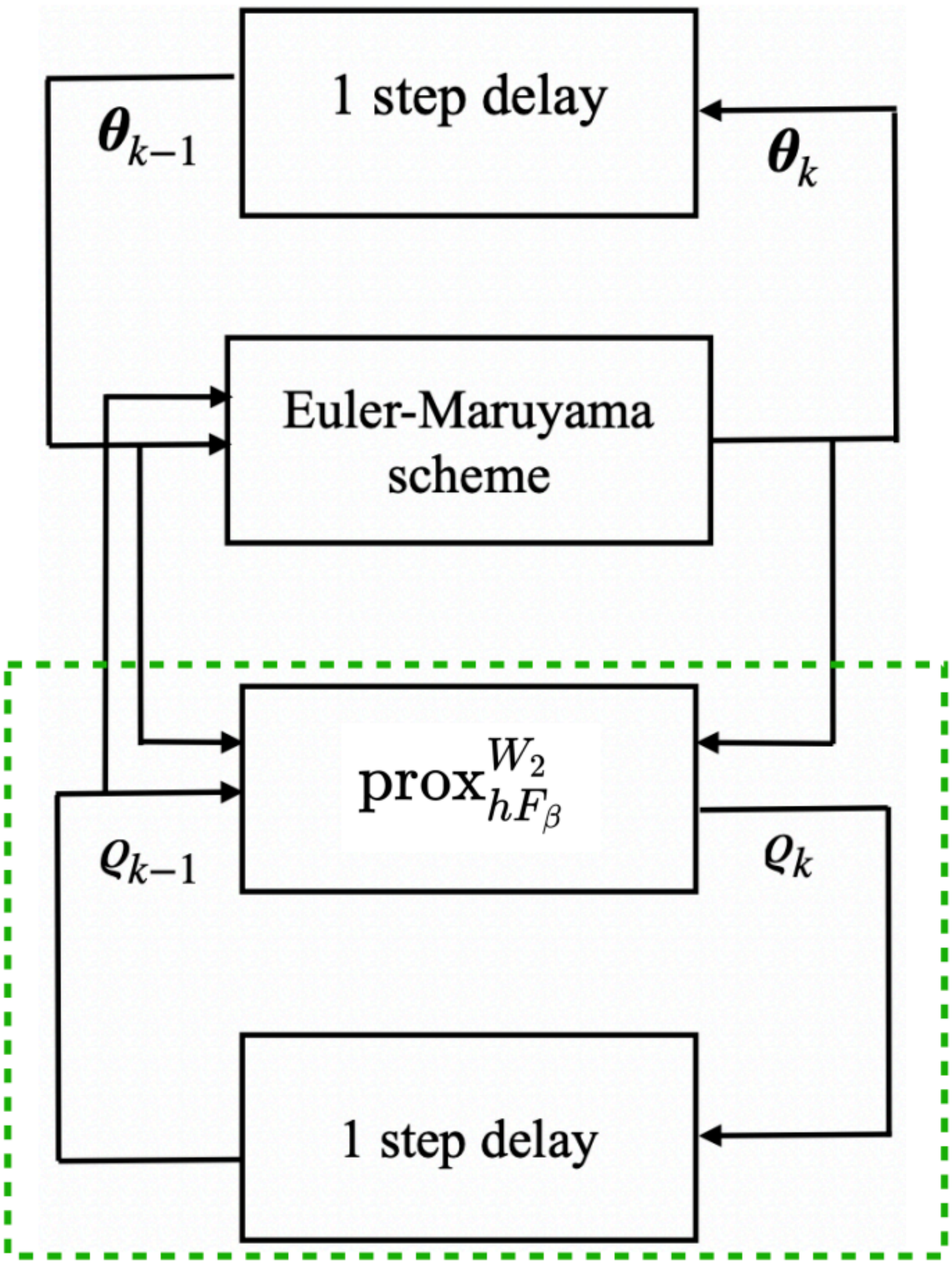
where $\varrho_{k-1}(\cdot) := \varrho(\cdot, t_{k-1})$

$$\varrho_0 \equiv \rho_0$$

Thm. 1:

As $h \to 0$, proximal updates converge to solution to PDE IVP.

ProxLearn Algorithm



Euler-Maruyama

$$\boldsymbol{\theta}_k^i = \boldsymbol{\theta}_{k-1}^i - h\nabla\left(V\left(\boldsymbol{\theta}_{k-1}^i\right) + \omega\left(\boldsymbol{\theta}_{k-1}^i\right)\right) + \sqrt{2\beta^{-1}}\left(\boldsymbol{\eta}_k^i - \boldsymbol{\eta}_{k-1}^i\right)$$

where $\quad \omega(\cdot) := \int U(\cdot, \tilde{\boldsymbol{\theta}})\varrho(\tilde{\boldsymbol{\theta}})\mathrm{d}\tilde{\boldsymbol{\theta}}$

and $\quad \boldsymbol{\eta}_{k-1}^i := \boldsymbol{\eta}^i(t = (k-1)h)$

Euler-Maruyama

$$\boldsymbol{\theta}_k^i = \boldsymbol{\theta}_{k-1}^i - h\nabla\left(V\left(\boldsymbol{\theta}_{k-1}^i\right) + \omega\left(\boldsymbol{\theta}_{k-1}^i\right)\right) + \sqrt{2\beta^{-1}}\left(\boldsymbol{\eta}_k^i - \boldsymbol{\eta}_{k-1}^i\right)$$

where $\omega(\cdot) := \int U(\cdot, \tilde{\boldsymbol{\theta}})\varrho(\tilde{\boldsymbol{\theta}})\mathrm{d}\tilde{\boldsymbol{\theta}}$

and $\boldsymbol{\eta}_{k-1}^i := \boldsymbol{\eta}^i(t = (k-1)h)$

Algorithm 2 Euler-Maruyama Algorithm

1: **procedure** EulerMaruyama($h, \beta, \boldsymbol{\Theta}_{k-1}, \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\varrho}_{k-1}$)
2: $\quad \boldsymbol{P}_{k-1} \leftarrow \boldsymbol{\Phi}(\boldsymbol{\Theta}_{k-1}, \boldsymbol{X})$ $\quad\triangleright$ Lines 2-4 construct the argument of the gradient in (35)
3: $\quad \boldsymbol{U}_{k-1} \leftarrow 1/n_{\text{data}}\boldsymbol{P}_{k-1}\boldsymbol{P}_{k-1}^\top$
4: $\quad \boldsymbol{u}_{k-1} \leftarrow \boldsymbol{U}_{k-1}\boldsymbol{\varrho}_{k-1}$
5: $\quad \boldsymbol{v}_{k-1} \leftarrow -2/n_{\text{data}}\boldsymbol{P}_{k-1}\boldsymbol{y}$
6: $\quad \boldsymbol{D} \leftarrow$ Backward $(\boldsymbol{u}_{k-1} + \boldsymbol{v}_{k-1})$ $\quad\triangleright$ Approximate the gradient of (35) using PyTorch library Backward (Paszke et al., 2017)
7: $\quad \boldsymbol{G} \leftarrow \sqrt{2h/\beta} \times \text{randn}_{N\times p}$
8: $\quad \boldsymbol{\Theta}_k \leftarrow \boldsymbol{\Theta}_{k-1} + h \times \boldsymbol{D} + \boldsymbol{G}$ $\quad\triangleright$ Complete the location update via (35)
9: **end procedure**

Derivation of ProxLearn

Proximal recursion (semi-implicit variant)

$$\varrho_k = \mathrm{prox}_{hF_\beta}^{W_2}(\varrho_{k-1}) := \underset{\varrho \in \mathcal{P}_2(\mathbb{R}^p)}{\mathrm{arginf}} \left\{ \frac{1}{2} \left(W_2(\varrho, \varrho_{k-1})\right)^2 + h\, \hat{F}_\beta(\varrho) \right\}$$

Derivation of ProxLearn

Discrete version of proximal recursion

$$\boldsymbol{\varrho}_k = \arg\min_{\boldsymbol{\varrho}} \left\{ \min_{\boldsymbol{M} \in \Pi(\boldsymbol{\varrho}_{k-1}, \boldsymbol{\varrho})} \frac{1}{2} \langle \boldsymbol{C}_k, \boldsymbol{M} \rangle + h \Big\langle \boldsymbol{v}_{k-1} + \boldsymbol{U}_{k-1} \boldsymbol{\varrho}_{k-1} + \beta^{-1} \log \boldsymbol{\varrho}, \boldsymbol{\varrho} \Big\rangle \right\}$$

where $\Pi(\boldsymbol{\varrho}_{k-1}, \boldsymbol{\varrho}) := \{\boldsymbol{M} \in \mathbb{R}^{N \times N} \mid \boldsymbol{M} \geq \boldsymbol{0} \text{ (elementwise)}, \boldsymbol{M}\boldsymbol{1} = \boldsymbol{\varrho}_{k-1},\ \boldsymbol{M}^\top \boldsymbol{1} = \boldsymbol{\varrho}\}$

Derivation of ProxLearn

Regularized discrete version of proximal recursion

$$\boldsymbol{\varrho}_k = \arg\min_{\boldsymbol{\varrho}} \left\{ \min_{\boldsymbol{M}\in\Pi(\boldsymbol{\varrho}_{k-1},\boldsymbol{\varrho})} \frac{1}{2}\langle \boldsymbol{C}_k, \boldsymbol{M}\rangle + \epsilon\langle \boldsymbol{M}, \log \boldsymbol{M}\rangle + h\big\langle \boldsymbol{v}_{k-1} + \boldsymbol{U}_{k-1}\boldsymbol{\varrho}_{k-1} + \beta^{-1}\log \boldsymbol{\varrho}, \boldsymbol{\varrho}\big\rangle \right\}$$

$$\text{where } \Pi(\boldsymbol{\varrho}_{k-1}, \boldsymbol{\varrho}) := \{\boldsymbol{M} \in \mathbb{R}^{N\times N} \mid \boldsymbol{M} \geq \boldsymbol{0} \text{ (elementwise)}, \boldsymbol{M}\boldsymbol{1} = \boldsymbol{\varrho}_{k-1},\ \boldsymbol{M}^\top\boldsymbol{1} = \boldsymbol{\varrho}\}$$

Derivation of ProxLearn

Regularized discrete version of proximal recursion

$$\boldsymbol{\varrho}_k = \arg\min_{\boldsymbol{\varrho}} \left\{ \min_{\boldsymbol{M}\in\Pi(\boldsymbol{\varrho}_{k-1},\boldsymbol{\varrho})} \frac{1}{2}\langle \boldsymbol{C}_k, \boldsymbol{M}\rangle + \epsilon\langle \boldsymbol{M}, \log \boldsymbol{M}\rangle + h\left\langle \boldsymbol{v}_{k-1} + \boldsymbol{U}_{k-1}\boldsymbol{\varrho}_{k-1} + \beta^{-1}\log \boldsymbol{\varrho}, \boldsymbol{\varrho}\right\rangle \right\}$$

where $\Pi(\boldsymbol{\varrho}_{k-1}, \boldsymbol{\varrho}) := \{\boldsymbol{M} \in \mathbb{R}^{N\times N} \mid \boldsymbol{M} \geq \boldsymbol{0} \text{ (elementwise)}, \boldsymbol{M}\boldsymbol{1} = \boldsymbol{\varrho}_{k-1},\ \boldsymbol{M}^\top\boldsymbol{1} = \boldsymbol{\varrho}\}$

Use Lagrange dual problem with Lagrange multipliers λ_0 and λ_1

Let:

$\boldsymbol{z} := \exp(\boldsymbol{\lambda}_1 h/\epsilon)$

$\boldsymbol{q} := \exp(\boldsymbol{\lambda}_0 h/\epsilon)$

$\boldsymbol{\Gamma}_k := \exp(-\boldsymbol{C}_k/2\epsilon)$

$\boldsymbol{\xi}_{k-1} := \exp(-\beta\boldsymbol{v}_{k-1} - \beta\boldsymbol{U}_{k-1}\boldsymbol{\varrho}_{k-1} - \boldsymbol{1})$

Proximal update

$$\boldsymbol{\varrho}_k = \boldsymbol{z} \odot \boldsymbol{\Gamma}_k^\top \boldsymbol{q}$$

Proximal algorithm

Algorithm 1 Proximal Algorithm

procedure ProxLearn($\boldsymbol{\varrho}_{k-1}, \boldsymbol{\Theta}_{k-1}, \beta, h, \varepsilon, N, \boldsymbol{X}, \boldsymbol{y}, \delta, L$)

  $\boldsymbol{v}_{k-1},\ \boldsymbol{U}_{k-1},\ \boldsymbol{\Theta}_k \leftarrow$ EulerMaruyama($h, \beta, \boldsymbol{\Theta}_{k-1}, \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\varrho}_{k-1}$) ▷ Update the location of the samples

  $\boldsymbol{C}_k(i,j) \leftarrow \left\| \boldsymbol{\theta}_k{}^i - \boldsymbol{\theta}_{k-1}^j \right\|_2^2$

  $\boldsymbol{\Gamma}_k \leftarrow \exp(-\boldsymbol{C}_k/2\varepsilon)$

  $\boldsymbol{\xi}_{k-1} \leftarrow \exp(-\beta \boldsymbol{v}_{k-1} - \beta \boldsymbol{U}_{k-1}\boldsymbol{\varrho}_{k-1} - \mathbf{1})$

  $\boldsymbol{z}_0 \leftarrow \mathrm{rand}_{N\times 1}$

  $\boldsymbol{z} \leftarrow [\boldsymbol{z}_0, \mathbf{0}_{N\times(L-1)}]$

  $\boldsymbol{q} \leftarrow [\boldsymbol{\varrho}_{k-1} \oslash (\boldsymbol{\Gamma_k z_0}), \mathbf{0}_{N\times(L-1)}]$

  $\ell = 1$

  while $\ell \le L$ **do**

    $\boldsymbol{z}(:,\ell+1) \leftarrow (\boldsymbol{\xi}_{k-1} \oslash (\boldsymbol{\Gamma}_k^\top \boldsymbol{q}(:,\ell)))^{\frac{1}{1+\beta\varepsilon/h}}$

    $\boldsymbol{q}(:,\ell+1) \leftarrow \boldsymbol{\varrho}_{k-1} \oslash (\boldsymbol{\Gamma}_k \boldsymbol{z}(:,\ell+1))$

    if $\|\boldsymbol{q}(:,\ell+1) - \boldsymbol{q}(:,\ell)\| < \delta$ and $\|\boldsymbol{z}(:,\ell+1) - \boldsymbol{z}(:,\ell)\| < \delta$ **then**

      Break

    else

      $\ell \leftarrow \ell + 1$

    end if

  end while

  return $\boldsymbol{\varrho}_k \leftarrow \boldsymbol{z}(:,\ell) \odot (\boldsymbol{\Gamma}_k^\top \boldsymbol{q}(:,\ell))$

end procedure

Case Study: Binary Classification on WDBC Data

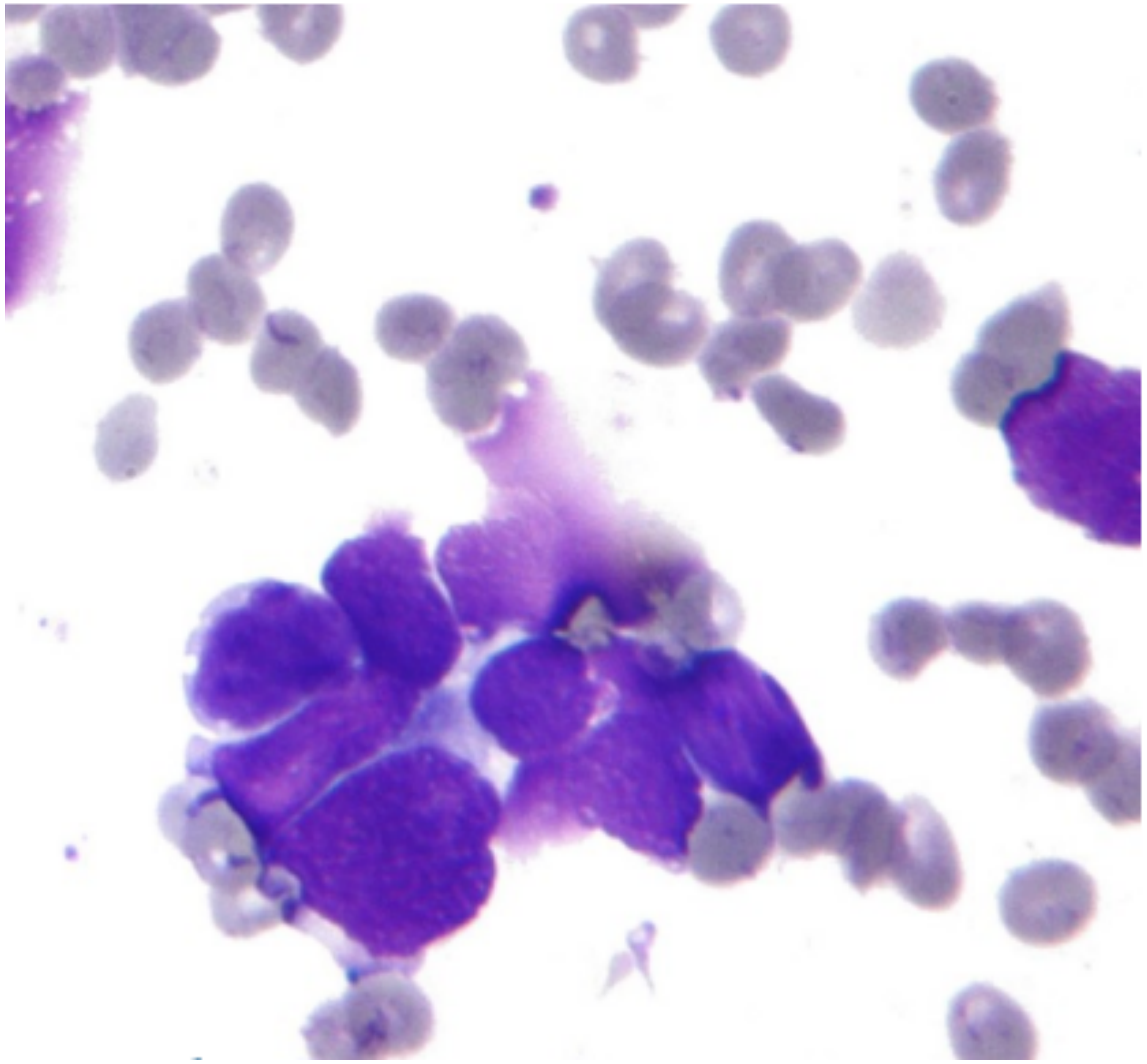
WDBC:

# of features:

$n_x = 30$

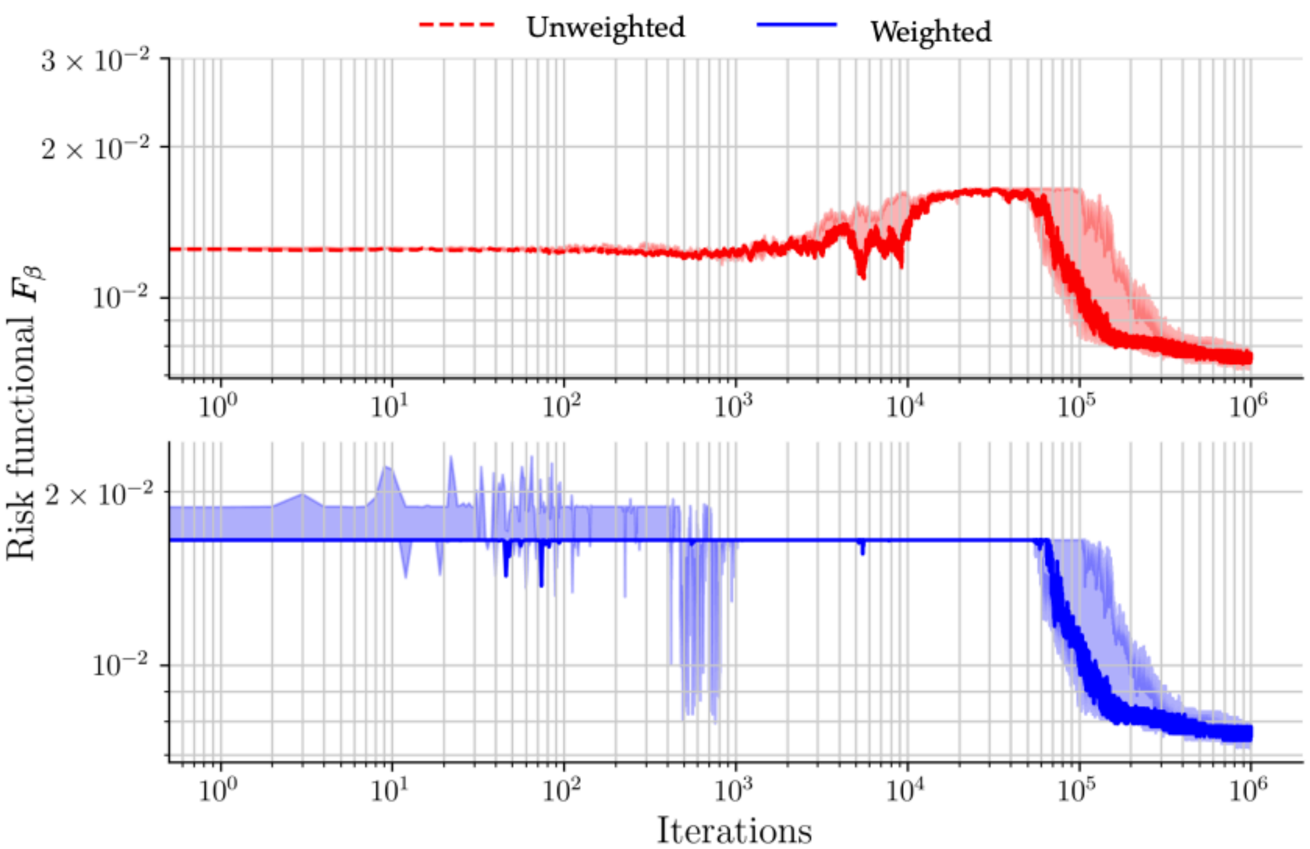
# of data points:

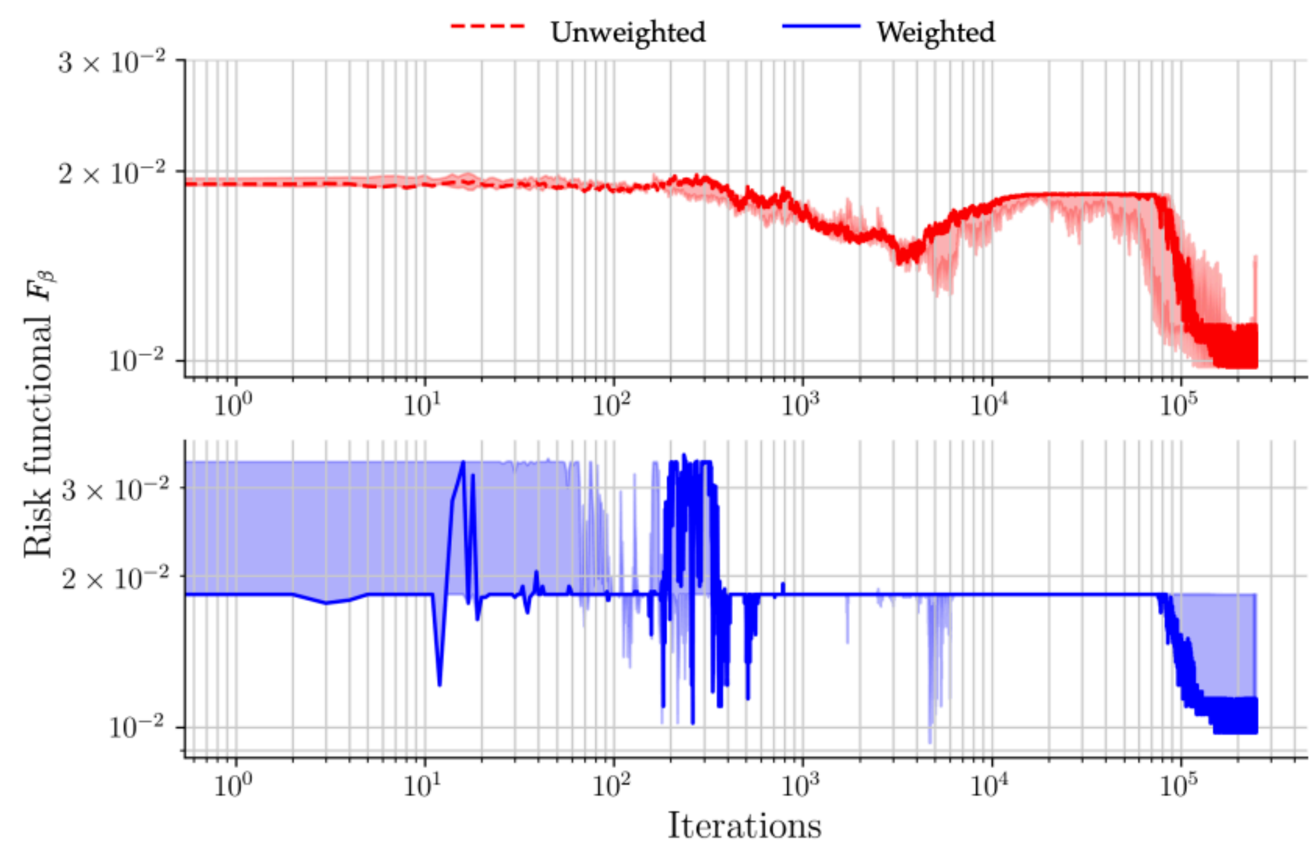
$n = 569$



Source: UCI machine learning repository, 2017, Available: http://archive.ics.uci.edu/ml/index.php

Case Study: Binary Classification on WDBC Data





β	Unweighted	Weighted
0.03	91.17%	92.35%
0.05	92.94%	92.94%
0.07	78.23%	92.94%

β	Unweighted	Weighted	Runtime (hr)
0.03	91.18%	91.18%	1.415
0.05	91.18%	92.94%	1.533
0.07	90.59%	91.76%	1.704

Case Study: Binary Classification

Comparison to Mokrov et al (2021) & Bonet et al (2022)

Banana:
\# of features: $n_x = 2$
\# of data points: $n = 5300$

Diabetes:
\# of features: $n_x = 8$
\# of data points: $n = 768$

Twonorm:
\# of features: $n_x = 20$
\# of data points: $n = 7400$

Dataset	JKO-ICNN	SWGF + RealNVP	ProxLearn, Weighted	ProxLearn, Unweighted
Banana	0.550 ± 10^{-2}	0.559 ± 10^{-2}	0.551 ± 10^{-2}	$0.535 \pm 5 \cdot 10^{-2}$
Diabetes	$0.777 \pm 7 \cdot 10^{-3}$	$0.778 \pm 2 \cdot 10^{-3}$	$0.736 \pm 2 \cdot 10^{-2}$	0.731 ± 10^{-2}
Twonorm	$0.981 \pm 2 \cdot 10^{-4}$	$0.981 \pm 6 \cdot 10^{-4}$	$0.972 \pm 2 \cdot 10^{-3}$	$0.972 \pm 2 \cdot 10^{-3}$

Case Study: Multi-Class Classification

Semeion Handwritten Digit Data Set

# of features:

$$n_x = 16 \times 16 = 256$$

# of data points:

$$n = 1593$$

Dua and Graff (2017) http://archive.ics.uci.edu/ml

Case Study: Multi-Class Classification

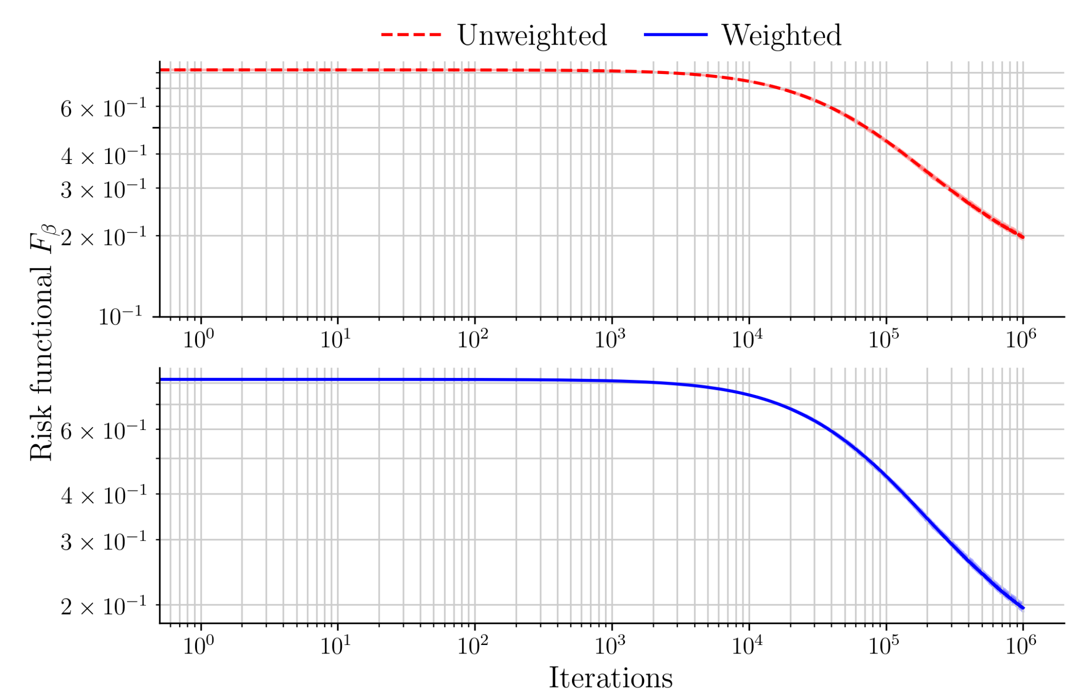
$$\begin{aligned}\boldsymbol{P}_{k-1}(j,i) &:= \boldsymbol{\Phi}(\boldsymbol{\theta}^j_{k-1}, \boldsymbol{X}(i,:), \boldsymbol{Y}(i,:)) \\ &:= \left\langle \operatorname{softmax}(\boldsymbol{X}(i,:)(\boldsymbol{\theta}^j_{k-1})^\top), (\boldsymbol{Y}(i,:))^\top \right\rangle\end{aligned}$$

Weighted

$$F_\beta \approx \frac{1}{n_{\text{test}}} \left\| \mathbf{1} - (\boldsymbol{P}^{\text{test}}_{k-1})^\top \boldsymbol{\varrho} \right\|_2^2$$

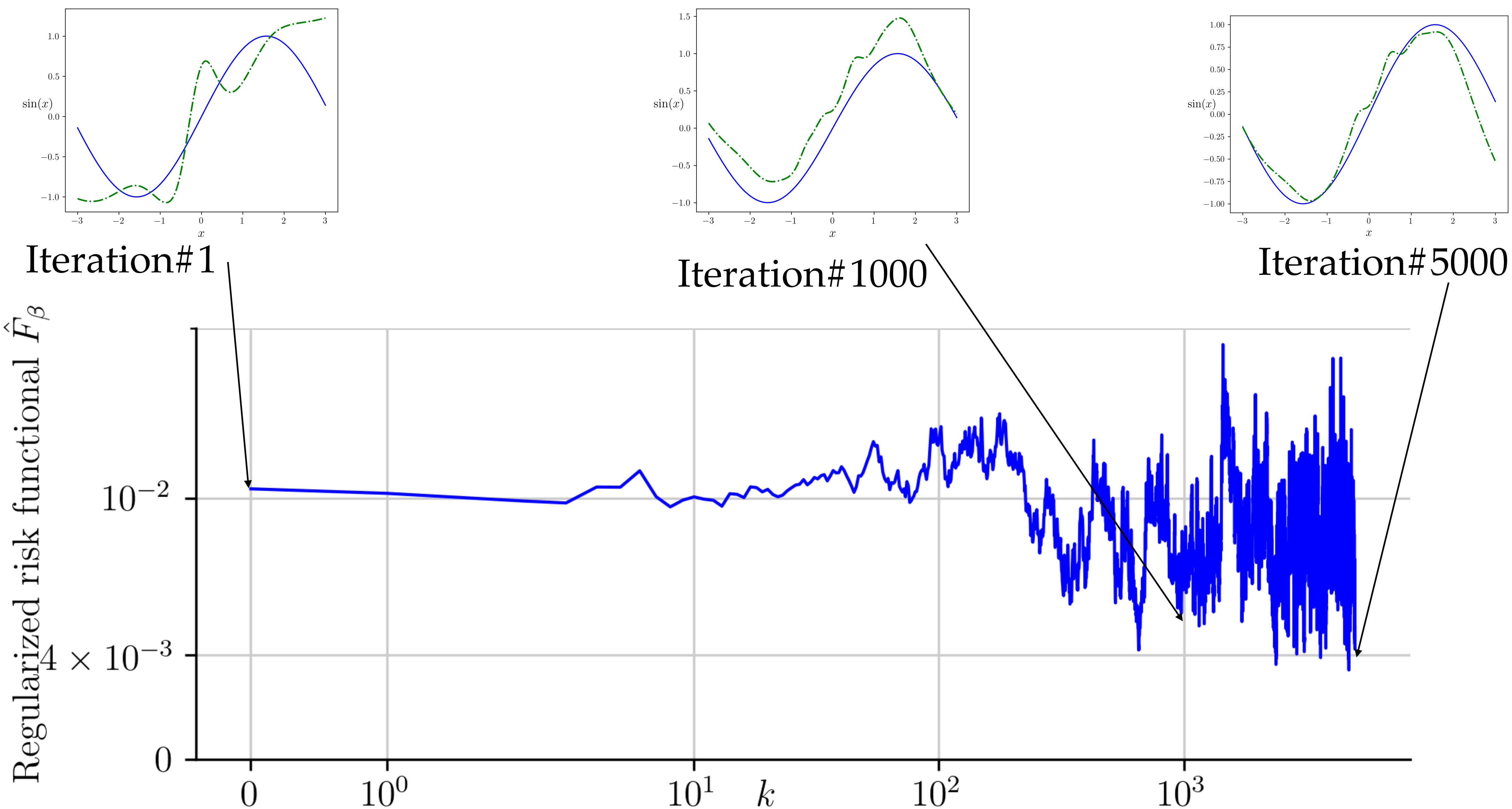
Unweighted

$$F_\beta \approx \frac{1}{n_{\text{test}}} \left\| \mathbf{1} - \frac{1}{N} (\boldsymbol{P}^{\text{test}}_{k-1})^\top \mathbf{1} \right\|_2^2$$

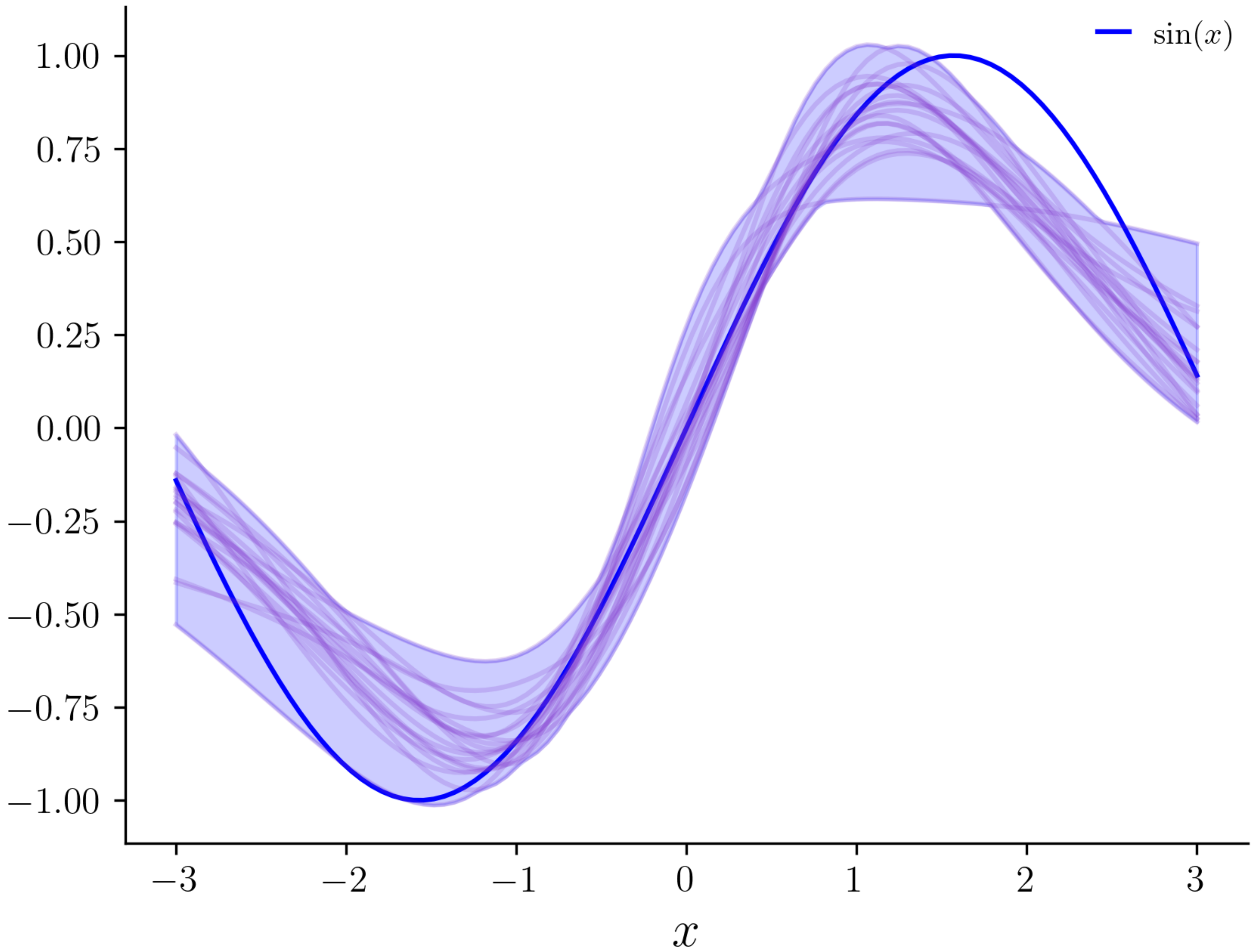


Learning a sinusoid

N	Final $\hat{F}_\beta$
500	0.01241931
700	0.01075817
1000	0.00806645
2000	0.00762518



Learning a sinusoid



Additional Avenues of Research

Multiple hidden layer setting

(*) Infinite width limit on one hidden layer; width of other hidden layers held constant

(*) Widths of all hidden layers go to infinity

Thank You

Acknowledgement: NSF **2112755**