Lattice Rules
Kernel Methods
Deep Neural Networks
and how to connect them

Frances Kuo

f.kuo@unsw.edu.au

School of Mathematics and Statistics, UNSW Sydney, Australia

Outline of my talk

- **Lattice rule** for high dimensional integration

  14 minutes

- **Lattice-based kernel interpolant** for high dimensional function approximation

  1 minute

- SIDE STEP: **Uncertainty Quantification** (**UQ**) and modeling random fields

  10 minutes

- **Lattice-based training for deep neural network** as function approximation

  15 minutes

Joint work with **Alexander Keller** (NVIDIA), **Dirk Nuyens** (KU Leuven), **Ian Sloan** (UNSW Sydney)

Thanks to my collaborators

NZ & AU

Austria

Belgium

Germany

Poland

Switzerland

UK

USA

Thanks to my collaborators

AU

Austria

Belgium

Ecuador

Finland

France

Germany

Switzerland

UK

USA

Shameless advertising

- **Uncertainty quantification (UQ) using periodic random variables** SINUM (2020)
  Vesa Kaarnioja (FU Berlin), Ian Sloan (UNSW Sydney)
- **Kernel method for function approximation in UQ** Numer. Math. (2021)
  Yoshihito Kazashi (Strathclyde), Fabio Nobile (EPFL), Vesa Kaarnioja (FU Berlin), Ian Sloan
- **Lattice algorithms for multivariate integration and approximation**
  Ronald Cools & Dirk Nuyens (KU Leuven), Ian Sloan MCOM75 (2020); MCOM (2021)
  Laurence Wilkes & Weiwen Mo (KU Leuven) J. Complexity (2023); Constr. App. (2024)
- **Smoothing by preintegration; CDF/PDF estimation; option pricing; log-normal PDE**
  Alexander Gilbert & Ian Sloan & Abirami Srikumar (UNSW Sydney)
  MCOM (2022); Springer (2022); SINUM (2022); SINUM (2025)
- **Parabolic PDE-constrained optimal control under uncertainty**
  Philipp Guth (JKU Linz), Vesa Kaarnioja & Claudia Schillings (FU Berlin), Ian Sloan
  Numer. Math. (2024); JUQ (2021)
- **Poisson equation on random domain** Numer. Math. (2024)
  Harri Hakula (Aalto), Helmut Harbrecht (Basel), Vesa Kaarnioja (FU Berlin), Ian Sloan
- **Helmholtz equation in random medium** submitted (2025)
  Ivan Graham & Euan Spence (Bath), Dirk Nuyens (KU Leuven), Ian Sloan
- **Regularity and tailored regularization of DNNs with application to parametric PDEs**
  Alexander Keller (Nvidia), Dirk Nuyens (KU Leuven), Ian Sloan submitted (2025)

Lattice Rules

Kernel Methods

(SIDE STEP: Parametric PDEs in UQ)

Deep Neural Networks

Monte Carlo v.s. Quasi-Monte Carlo methods

$$\boxed{\int_{[0,1]^s} F(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \approx \frac{1}{N}\sum_{k=0}^{N-1} F(\boldsymbol{t}_k)}$$

Monte Carlo method (MC)

$\boldsymbol{t}_k$ random uniform

$N^{-1/2}$ convergence

Order of variables irrelevant

Quasi-Monte Carlo methods (QMC)

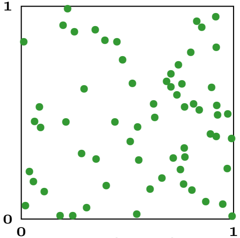
$\boldsymbol{t}_k$ deterministic

Close to N^{-1} convergence or better

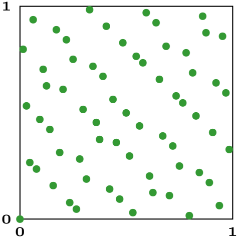
Better for earlier variables and lower-order projections

Order of variables very important

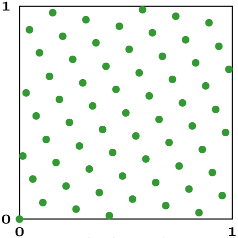
Good for integrands with *low effective dimension*



64 random points



64 Sobol′ points



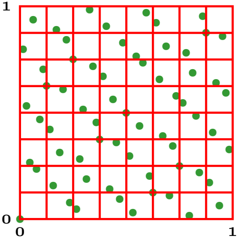
64 lattice points

Quasi-Monte Carlo methods (QMC)

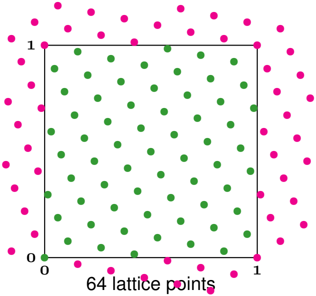
$$\boxed{\int_{[0,1]^s} F(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \approx \frac{1}{N}\sum_{k=0}^{N-1} F(\boldsymbol{t}_k)}$$

Niederreiter (1992)
Sloan, Joe (1994)
Dick, Pillichshammer (2010)
Dick, Kuo, Sloan (2013)
Dick, Kritzer, Pillichshammer (2022)

- How do we choose good QMC points $\boldsymbol{t}_k$?
- How would the error behave w.r.t. N?
- How would the error behave w.r.t. s?



64 Sobol′ points



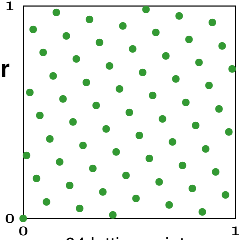
64 lattice points

Lifting the curse of dimensionality

- How would the error behave with respect to the dimension s?
  - Want **strong tractability**, i.e., error bounded independently of s
  - Work in **weighted** function space setting (“effective dimension”)

Sloan, Woźniakowski (1998), Novak, Woźniakowski (2008, 2010, 2012)

- How would the error behave with respect to the number of points N?
  - Want **higher order** convergence rate when the integrand is smooth
- How do we choose good QMC points t_k?
  - Use ***fast* component-by-component construction**
  - For lattice rules, we need just one **generating vector**
- How do we estimate the error in practice?
  - Use **randomization**
  - For lattice rules, we use **random shifts**



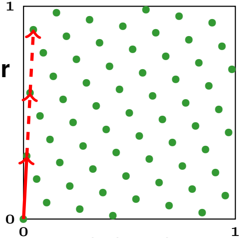
64 lattice points

Lifting the curse of dimensionality

- How would the error behave with respect to the dimension s?
  - Want **strong tractability**, i.e., error bounded independently of s
  - Work in **weighted** function space setting (“effective dimension”)

Sloan, Woźniakowski (1998), Novak, Woźniakowski (2008, 2010, 2012)

- How would the error behave with respect to the number of points N?
  - Want **higher order** convergence rate when the integrand is smooth
- How do we choose good QMC points t_k?
  - Use ***fast* component-by-component construction**
  - For lattice rules, we need just one **generating vector**
- How do we estimate the error in practice?
  - Use **randomization**
  - For lattice rules, we use **random shifts**



64 lattice points

Lifting the curse of dimensionality

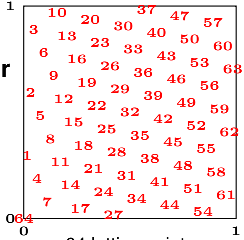
- How would the error behave with respect to the dimension s?
  - Want **strong tractability**, i.e., error bounded independently of s
  - Work in **weighted** function space setting ("effective dimension")

Sloan, Woźniakowski (1998), Novak, Woźniakowski (2008, 2010, 2012)

- How would the error behave with respect to the number of points N?
  - Want **higher order** convergence rate when the integrand is smooth

- How do we choose good QMC points $\boldsymbol{t}_k$?
  - Use ***fast* component-by-component construction**
  - For lattice rules, we need just one **generating vector**

- How do we estimate the error in practice?
  - Use **randomization**
  - For lattice rules, we use **random shifts**



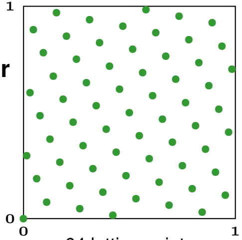
64 lattice points

Lifting the curse of dimensionality

- How would the error behave with respect to the dimension s?
  - Want **strong tractability**, i.e., error bounded independently of s
  - Work in **weighted** function space setting ("effective dimension")

Sloan, Woźniakowski (1998), Novak, Woźniakowski (2008, 2010, 2012)

- How would the error behave with respect to the number of points N?
  - Want **higher order** convergence rate when the integrand is smooth
- How do we choose good QMC points $\boldsymbol{t}_k$?
  - Use ***fast* component-by-component construction**
  - For lattice rules, we need just one **generating vector**
- How do we estimate the error in practice?
  - Use **randomization**
  - For lattice rules, we use **random shifts**



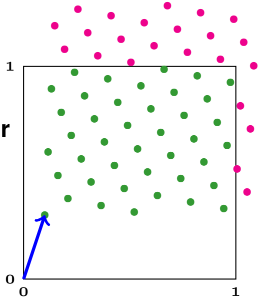
64 lattice points

Lifting the curse of dimensionality

- How would the error behave with respect to the dimension s?
  - Want **strong tractability**, i.e., error bounded independently of s
  - Work in **weighted** function space setting ("effective dimension")

Sloan, Woźniakowski (1998), Novak, Woźniakowski (2008, 2010, 2012)

- How would the error behave with respect to the number of points N?
  - Want **higher order** convergence rate when the integrand is smooth
- How do we choose good QMC points $\boldsymbol{t}_k$?
  - Use ***fast* component-by-component construction**
  - For lattice rules, we need just one **generating vector**
- How do we estimate the error in practice?
  - Use **randomization**
  - For lattice rules, we use **random shifts**



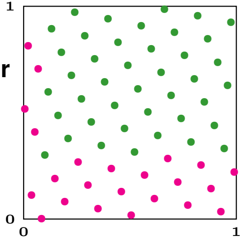
64 shifted lattice points

Lifting the curse of dimensionality

- How would the error behave with respect to the dimension s?
  - Want **strong tractability**, i.e., error bounded independently of s
  - Work in **weighted** function space setting ("effective dimension")

Sloan, Woźniakowski (1998), Novak, Woźniakowski (2008, 2010, 2012)

- How would the error behave with respect to the number of points N?
  - Want **higher order** convergence rate when the integrand is smooth
- How do we choose good QMC points $\boldsymbol{t}_k$?
  - Use ***fast* component-by-component construction**
  - For lattice rules, we need just one **generating vector**
- How do we estimate the error in practice?
  - Use **randomization**
  - For lattice rules, we use **random shifts**



64 shifted lattice points

Component-by-component construction

- Want to find $\boldsymbol{z}$ for which some error criterion is as small as possible?
  - Exhaustive search is practically impossible - too many choices!
- **Component-by-component construction (CBC)**

  Korobov (1959), Sloan, Reztsov (2002), Sloan, Kuo, Joe (2022), ...

  1. Set $z_1 = 1$
  2. With z_1 fixed, choose z_2 to minimize the error criterion in 2 dimensions
  3. With z_1, z_2 fixed, choose z_3 to minimize the error criterion in 3 dimensions
  4. etc.

  Kuo (2003), Dick (2004), ...

- Optimal rate of convergence $\mathcal{O}(N^{-1+\delta})$ in "weighted Sobolev space",
  $\mathcal{O}(N^{-\alpha+\delta})$ in "weighted Korobov space",
  independently of s under an appropriate condition on the weights
  - Averaging argument: there is always one choice as good as average!
- Cost of algorithm for "product weights" is $\mathcal{O}(s\, N \log N)$ using FFT

  Nuyens, Cools (2006), ...

- Extensible/embedded variants

  Hickernell, Hong, L'Ecuyer, Lemieux (2000), Hickernell, Niederreiter (2003), Cools, Kuo, Nuyens (2006), Dick, Pillichshammer, Waterhouse (2007)

How to apply QMC theory (most important slide)

- Common features among “modern” QMC theoretical settings:
  - Separation in error bound

    $$\boxed{\text{(rms) } \textbf{QMC error} \le \text{(rms) } \textbf{worst case error}_{\boldsymbol{\gamma}} \times \textbf{norm of integrand}_{\boldsymbol{\gamma}}}$$

  - Weighted spaces
  - Pairing of QMC rule with function space
  - Optimal rate of convergence
  - Rate and constant independent of dimension
  - **Fast component-by-component (CBC) construction**
  - Extensible or embedded rules

How to apply QMC theory (most important slide)

- Common features among "modern" QMC theoretical settings:
  - Separation in error bound

    $\boxed{\text{(rms) } \textbf{QMC error} \le \text{(rms) } \textbf{worst case error}_{\boldsymbol{\gamma}} \times \textbf{norm of integrand}_{\boldsymbol{\gamma}}}$

  - Weighted spaces
  - Pairing of QMC rule with function space
  - Optimal rate of convergence
  - Rate and constant independent of dimension
  - **Fast component-by-component (CBC) construction**
  - Extensible or embedded rules
- Application of QMC theory
  - **Estimate the norm** (critical step)
  - **Choose the weights**
  - Weights as input to the CBC construction

Two (out of many) theoretical settings for lattice rules

$\boxed{\text{(rms) QMC error} \le \text{(rms) worst case error}_{\boldsymbol{\gamma}} \times \text{norm of integrand}_{\boldsymbol{\gamma}}}$

(a) Randomly-shifted lattice rules + weighted Sobolev space of smoothness 1

$$\|F\|^2_{\mathcal{W}_{1,\boldsymbol{\gamma}}} := \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{1}{\gamma_{\mathfrak{u}}} \int_{[0,1]^{|\mathfrak{u}|}} \bigg| \int_{[0,1]^{s-|\mathfrak{u}|}} \partial^{\mathbf{1}_{\mathfrak{u}}} F(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}_{-\mathfrak{u}} \bigg|^2 \mathrm{d}\boldsymbol{y}_{\mathfrak{u}}$$

Mixed **first** derivatives are square integrable

2^s subsets

Small "weight" $\gamma_{\mathfrak{u}}$ means that F depends weakly on the variables $\boldsymbol{y}_{\mathfrak{u}}$

(b) Lattice rules + (periodic) weighted Korobov space of smoothness α

$$\|F\|^2_{\mathcal{W}_{\alpha,\boldsymbol{\gamma}}} := \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{1}{\gamma_{\mathfrak{u}}} \int_{[0,1]^{|\mathfrak{u}|}} \bigg| \int_{[0,1]^{s-|\mathfrak{u}|}} \partial^{\boldsymbol{\alpha}_{\mathfrak{u}}} F(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}_{-\mathfrak{u}} \bigg|^2 \mathrm{d}\boldsymbol{y}_{\mathfrak{u}}$$

Two (out of many) theoretical settings for lattice rules

(rms) **QMC error** $\le$ (rms) **worst case error**$_{\boldsymbol{\gamma}}$ $\times$ **norm of integrand**$_{\boldsymbol{\gamma}}$

(a) Randomly-shifted lattice rules + weighted Sobolev space of smoothness 1

$$\|F\|^2_{\mathcal{W}_{1,\boldsymbol{\gamma}}} := \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{1}{\gamma_{\mathfrak{u}}} \int_{[0,1]^{|\mathfrak{u}|}} \left| \int_{[0,1]^{s-|\mathfrak{u}|}} \partial^{\mathbf{1}_{\mathfrak{u}}} F(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y}_{-\mathfrak{u}} \right|^2 \mathrm{d}\boldsymbol{y}_{\mathfrak{u}} \le \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{B_{\mathfrak{u}}}{\gamma_{\mathfrak{u}}}$$

Mixed **first** derivatives are square integrable

2^s subsets

Small "weight" $\gamma_{\mathfrak{u}}$ means that F depends weakly on the variables $\boldsymbol{y}_{\mathfrak{u}}$

(b) Lattice rules + (periodic) weighted Korobov space of smoothness α

$$\|F\|^2_{\mathcal{W}_{\alpha,\boldsymbol{\gamma}}} := \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{1}{\gamma_{\mathfrak{u}}} \int_{[0,1]^{|\mathfrak{u}|}} \left| \int_{[0,1]^{s-|\mathfrak{u}|}} \partial^{\boldsymbol{\alpha}_{\mathfrak{u}}} F(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y}_{-\mathfrak{u}} \right|^2 \mathrm{d}\boldsymbol{y}_{\mathfrak{u}} \le \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{B_{\mathfrak{u}}}{\gamma_{\mathfrak{u}}}$$

Two (out of many) theoretical settings for lattice rules

$\boxed{\text{(rms) QMC error} \le \text{(rms) worst case error}_{\boldsymbol{\gamma}} \times \text{norm of integrand}_{\boldsymbol{\gamma}}}$

(a) Randomly-shifted lattice rules + weighted Sobolev space of smoothness 1

$$\|F\|^2_{\mathcal{W}_{1,\boldsymbol{\gamma}}} := \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{1}{\gamma_{\mathfrak{u}}} \int_{[0,1]^{|\mathfrak{u}|}} \left| \int_{[0,1]^{s-|\mathfrak{u}|}} \partial^{\mathbf{1}_{\mathfrak{u}}} F(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y}_{-\mathfrak{u}} \right|^2 \mathrm{d}\boldsymbol{y}_{\mathfrak{u}} \le \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{B_{\mathfrak{u}}}{\gamma_{\mathfrak{u}}}$$

Mixed **first** derivatives are square integrable

2^s subsets

Small "weight" $\gamma_{\mathfrak{u}}$ means that F depends weakly on the variables $\boldsymbol{y}_{\mathfrak{u}}$

$$\text{rms } e^{\text{wor-int}} \le \left(\frac{2}{N} \sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}}^{\lambda} A_{\mathfrak{u}}\right)^{\frac{1}{2\lambda}} \quad \forall\, \lambda \in (\tfrac{1}{2}, 1]$$

$\boxed{\text{close to } \mathcal{O}(N^{-1})}$

(b) Lattice rules + (periodic) weighted Korobov space of smoothness α

$$\|F\|^2_{\mathcal{W}_{\alpha,\boldsymbol{\gamma}}} := \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{1}{\gamma_{\mathfrak{u}}} \int_{[0,1]^{|\mathfrak{u}|}} \left| \int_{[0,1]^{s-|\mathfrak{u}|}} \partial^{\boldsymbol{\alpha}_{\mathfrak{u}}} F(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y}_{-\mathfrak{u}} \right|^2 \mathrm{d}\boldsymbol{y}_{\mathfrak{u}} \le \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{B_{\mathfrak{u}}}{\gamma_{\mathfrak{u}}}$$

$$e^{\text{wor-int}} \le \left(\frac{2}{N} \sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}}^{\lambda} A_{\mathfrak{u}}\right)^{\frac{1}{2\lambda}} \quad \forall\, \lambda \in (\tfrac{1}{2\alpha}, 1]$$

$\boxed{\text{close to } \mathcal{O}(N^{-\alpha})}$

Two (out of many) theoretical settings for lattice rules

(rms) **QMC error** $\lesssim$ (rms) **worst case error**$_{\boldsymbol{\gamma}}$ $\times$ **norm of integrand**$_{\boldsymbol{\gamma}}$

(a) Randomly-shifted lattice rules + weighted Sobolev space of smoothness 1

$$\|F\|^2_{\mathcal{W}_{1,\boldsymbol{\gamma}}} := \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{1}{\gamma_{\mathfrak{u}}} \int_{[0,1]^{|\mathfrak{u}|}} \left| \int_{[0,1]^{s-|\mathfrak{u}|}} \partial^{\mathbf{1}_{\mathfrak{u}}} F(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}_{-\mathfrak{u}} \right|^2 \mathrm{d}\boldsymbol{y}_{\mathfrak{u}} \le \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{B_{\mathfrak{u}}}{\gamma_{\mathfrak{u}}}$$

Mixed **first** derivatives are square integrable

2^s subsets

Small "weight" $\gamma_{\mathfrak{u}}$ means that F depends weakly on the variables $\boldsymbol{y}_{\mathfrak{u}}$

$$\text{rms } e^{\text{wor-int}} \le \left(\frac{2}{N} \sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}}^{\lambda} A_{\mathfrak{u}} \right)^{\frac{1}{2\lambda}} \quad \forall\, \lambda \in (\tfrac{1}{2}, 1]$$

close to $\mathcal{O}(N^{-1})$

(b) Lattice rules + (periodic) weighted Korobov space of smoothness α

$$\|F\|^2_{\mathcal{W}_{\alpha,\boldsymbol{\gamma}}} := \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{1}{\gamma_{\mathfrak{u}}} \int_{[0,1]^{|\mathfrak{u}|}} \left| \int_{[0,1]^{s-|\mathfrak{u}|}} \partial^{\boldsymbol{\alpha}_{\mathfrak{u}}} F(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}_{-\mathfrak{u}} \right|^2 \mathrm{d}\boldsymbol{y}_{\mathfrak{u}} \le \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{B_{\mathfrak{u}}}{\gamma_{\mathfrak{u}}}$$

$$e^{\text{wor-int}} \le \left(\frac{2}{N} \sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}}^{\lambda} A_{\mathfrak{u}} \right)^{\frac{1}{2\lambda}} \quad \forall\, \lambda \in (\tfrac{1}{2\alpha}, 1]$$

close to $\mathcal{O}(N^{-\alpha})$

Choose weights for CBC construction

$$\text{minimize} \quad \left(\sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}}^{\lambda} A_{\mathfrak{u}} \right)^{\frac{1}{2\lambda}} \left(\sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{B_{\mathfrak{u}}}{\gamma_{\mathfrak{u}}} \right)^{\frac{1}{2}} \quad \Longrightarrow \quad \gamma_{\mathfrak{u}} = \left(\frac{B_{\mathfrak{u}}}{A_{\mathfrak{u}}} \right)^{\frac{1}{1+\lambda}}$$

Lattice Rules

Kernel Methods

(SIDE STEP: Parametric PDEs in UQ)

Deep Neural Networks

From integration to function approximation

- **Integration over the unit cube**

$$\int_{[0,1]^s} F(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \approx \frac{1}{N}\sum_{k=0}^{N-1} F(\boldsymbol{t}_k)$$

- **Integration over the Euclidean space**

$$\cdots = \int_{\mathbb{R}^s} F(\boldsymbol{y}) \prod_{i=1}^{s} \phi(y_i)\,\mathrm{d}\boldsymbol{y} = \int_{[0,1]^s} F(\boldsymbol{\Phi}^{-1}(\boldsymbol{w}))\,\mathrm{d}\boldsymbol{w} \approx \frac{1}{N}\sum_{k=0}^{N-1} F(\boldsymbol{\Phi}^{-1}(\boldsymbol{t}_k))$$

From integration to function approximation

- **Integration over the unit cube**

$$\int_{[0,1]^s} F(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \approx \frac{1}{N}\sum_{k=0}^{N-1} F(\boldsymbol{t}_k)$$

- **Integration over the Euclidean space**

$$\cdots = \int_{\mathbb{R}^s} F(\boldsymbol{y}) \prod_{i=1}^{s} \phi(y_i)\,\mathrm{d}\boldsymbol{y} = \int_{[0,1]^s} F(\boldsymbol{\Phi}^{-1}(\boldsymbol{w}))\,\mathrm{d}\boldsymbol{w} \approx \frac{1}{N}\sum_{k=0}^{N-1} F(\boldsymbol{\Phi}^{-1}(\boldsymbol{t}_k))$$

- **Function approximation over the unit cube**

$$\int_{[0,1]^s} F(\boldsymbol{y}) e^{-2\pi \mathrm{i} \boldsymbol{h}\cdot\boldsymbol{y}}\,\mathrm{d}\boldsymbol{y} \approx \tfrac{1}{N}\textstyle\sum_{k=0}^{N-1} F(\boldsymbol{t}_k) e^{-2\pi \mathrm{i} \boldsymbol{h}\cdot\boldsymbol{t}_k}$$

  - Truncated trig. series

$$F(\boldsymbol{y}) = \sum_{\boldsymbol{h}\in\mathbb{Z}^s} \widehat{F}_{\boldsymbol{h}}\, e^{2\pi \mathrm{i} \boldsymbol{h}\cdot\boldsymbol{y}} \approx \underbrace{\sum_{\boldsymbol{h}\in\mathcal{A}_s} \widehat{F}^a_{\boldsymbol{h}}\, e^{2\pi \mathrm{i} \boldsymbol{h}\cdot\boldsymbol{y}}}_{=: A_N^{\mathrm{trig}}(F)(\boldsymbol{y})}$$

From integration to function approximation

- **Integration over the unit cube**

$$\int_{[0,1]^s} F(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \approx \frac{1}{N}\sum_{k=0}^{N-1} F(\boldsymbol{t}_k)$$

- **Integration over the Euclidean space**

$$\cdots = \int_{\mathbb{R}^s} F(\boldsymbol{y}) \prod_{i=1}^{s} \phi(y_i)\,\mathrm{d}\boldsymbol{y} = \int_{[0,1]^s} F(\Phi^{-1}(\boldsymbol{w}))\,\mathrm{d}\boldsymbol{w} \approx \frac{1}{N}\sum_{k=0}^{N-1} F(\Phi^{-1}(\boldsymbol{t}_k))$$

- **Function approximation over the unit cube**

$$\int_{[0,1]^s} F(\boldsymbol{y}) e^{-2\pi\mathrm{i}\boldsymbol{h}\cdot\boldsymbol{y}}\,\mathrm{d}\boldsymbol{y} \approx \tfrac{1}{N}\sum_{k=0}^{N-1} F(\boldsymbol{t}_k) e^{-2\pi\mathrm{i}\boldsymbol{h}\cdot\boldsymbol{t}_k}$$

  - Truncated trig. series $\quad F(\boldsymbol{y}) = \sum_{\boldsymbol{h}\in\mathbb{Z}^s} \widehat{F}_{\boldsymbol{h}}\, e^{2\pi\mathrm{i}\boldsymbol{h}\cdot\boldsymbol{y}} \approx \underbrace{\sum_{\boldsymbol{h}\in\mathcal{A}_s} \widehat{F}^a_{\boldsymbol{h}}\, e^{2\pi\mathrm{i}\boldsymbol{h}\cdot\boldsymbol{y}}}_{=:A_N^{\mathrm{trig}}(F)(\boldsymbol{y})}$

  - Kernel method $\quad F(\boldsymbol{y}) \approx \underbrace{\sum_{k=0}^{N-1} a_k K(\boldsymbol{t}_k, \boldsymbol{y})}_{=:A_N^{\mathrm{ker}}(F)(\boldsymbol{y})}$ so that $F(\boldsymbol{t}_\ell) = A_N^{\mathrm{ker}}(F)(\boldsymbol{t}_\ell)\ \forall \ell$

Lattice Rules

Kernel Methods

(SIDE STEP: Parametric PDEs in UQ)

Deep Neural Networks

Eg.1 Sound-soft plane-wave scattering

- **Helmholtz equation**
  - Frequency domain: wavenumber k

    $F(\boldsymbol{x}, t) = \exp(\mathrm{i}\, k\, t)\, g(\boldsymbol{x}) \implies U(\boldsymbol{x}, t) = \exp(\mathrm{i}\, k\, t)\, u(\boldsymbol{x})$

$$\boxed{-\nabla \cdot (A \nabla u) - k^2\, a\, u = g}$$

plus boundary conditions

- **Uncertainty Quantification (UQ)**
  - Random coefficients: $A(\boldsymbol{x}, \omega)$ and $a(\boldsymbol{x}, \omega)$
  - Quantities of interest: expected values with respect to ω
  - **Forward problem**: how do fluctuations in A and a affect u
  - **Inverse problem**: given observations of u, infer distribution of A and a

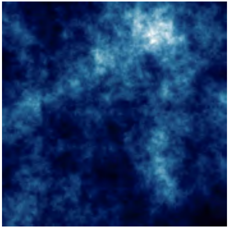
Graham, Kuo, Nuyens, Spence, Sloan (submitted 2025)

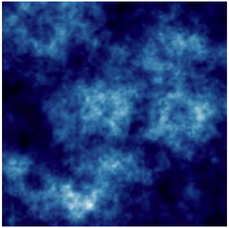
Eg.2 Uncertainty in groundwater flow

- **Risk analysis of radwaste disposal or CO_2 sequestration**
  - Darcy's law
  - Mass conservation law

$$\begin{cases} \boldsymbol{q}(\boldsymbol{x}) + a(\boldsymbol{x},\omega)\,\nabla p(\boldsymbol{x}) = f(\boldsymbol{x}) \\ \nabla \cdot \boldsymbol{q}(\boldsymbol{x}) = 0 \end{cases}$$

in $D \subset \mathbb{R}^d$, $d = 1, 2, 3$





  - Uncertainty in $a(\boldsymbol{x},\omega)$ leads to uncertainty in $\boldsymbol{q}(\boldsymbol{x},\omega)$ and $p(\boldsymbol{x},\omega)$

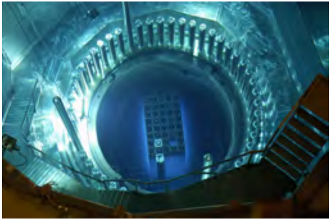
Graham, Kuo, Nuyens, Scheichl, Sloan (J. Comput. Physics 2011), . . .

Eg.3 Criticality problem for nuclear reactors

- **PDE eigenvalue problem**

$$\nabla \cdot (\underbrace{a(\boldsymbol{x},\omega)}_{\text{diffusion}} \nabla u(\boldsymbol{x})) + \underbrace{b(\boldsymbol{x},\omega)}_{\text{absorption}} u(\boldsymbol{x}) = \lambda \underbrace{c(\boldsymbol{x},\omega)}_{\text{fission}} u(\boldsymbol{x})$$

  - Smallest eigenvalue λ_1 measures *criticality* of reactor
  - Eigenfunction $u_1(\boldsymbol{x})$ is the *neutron flux* at the point $\boldsymbol{x}$
    - $\lambda_1 \approx 1 \implies$ operating efficiently
    - $\lambda_1 > 1 \implies$ not self-sustaining
    - $\lambda_1 < 1 \implies$ supercritical



https://www.youtube.com/watch?v=xIDytUCRtTA

Gilbert, Graham, Kuo, Scheichl, Sloan (Numer. Math. 2019), . . .

Eg.4 Optimal control under uncertainty

- **PDE-constrained robust optimal control**: steer u toward target g

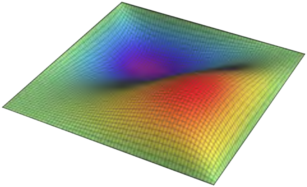
$$\min_{z\in\mathcal{Z}} J(u,z), \quad J(u,z) = \frac{1}{2}\int_U \|u(\cdot,\boldsymbol{y}) - g\|^2\,\mathrm{d}\boldsymbol{y} + \frac{\alpha}{2}\|z\|^2$$

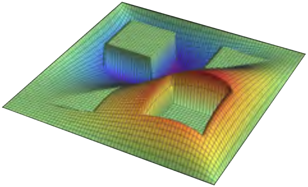
subject to

$$-\nabla\cdot(a(\boldsymbol{x},\boldsymbol{y})\nabla u(\boldsymbol{x},\boldsymbol{y})) = z(\boldsymbol{x}) \quad \boldsymbol{x}\in D,\ \boldsymbol{y}\in U$$

$$u(\boldsymbol{x},\boldsymbol{y}) = 0 \quad \boldsymbol{x}\in\partial D,\ \boldsymbol{y}\in U$$

$$z\in\mathcal{Z} = \{z\in L^2(D) : z_{\min}(\boldsymbol{x}) \le z(\boldsymbol{x}) \le z_{\max}(\boldsymbol{x}) \quad \text{a.e. in } D\}$$



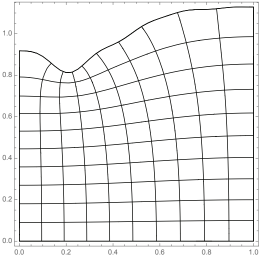


Guth, Kaarnioja, Kuo, Schillings, Sloan (SIAM/ASA J. Uncertain. Quantif. 2021; Numer. Math. 2024), . . .

Eg.5 Poisson equation on random domain

- **Poisson equation on random domains**

$$\Delta u(\boldsymbol{x},\omega) = f(\boldsymbol{x}) \quad \boldsymbol{x} \in D(\omega)$$
$$u(\boldsymbol{x},\omega) = 0 \qquad\quad \boldsymbol{x} \in \partial D(\omega)$$



- Domain mapping method
  - Reference domain D_{ref} (Lipschitz)
  - Admissible domain $D(\boldsymbol{y}) = V(D_{\mathrm{ref}}, \boldsymbol{y})$
  - Perturbation field $V(\boldsymbol{x},\boldsymbol{y})$ and Jacobian matrix $J(\boldsymbol{x},\boldsymbol{y})$
  - Variational formulation

$$\int_{D(\boldsymbol{y})} \nabla u(\boldsymbol{x},\boldsymbol{y}) \cdot \nabla v(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\boldsymbol{x} = \int_{D(\boldsymbol{y})} f(\boldsymbol{x})\, v(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\boldsymbol{x} \quad \forall\, v \in H_0^1(D(\boldsymbol{y}))$$

$$\int_{D_{\mathrm{ref}}} (A(\boldsymbol{x},\boldsymbol{y}) \nabla \widehat{u}(\boldsymbol{x},\boldsymbol{y})) \cdot \nabla \widehat{v}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_{D_{\mathrm{ref}}} f_{\mathrm{ref}}(\boldsymbol{x},\boldsymbol{y})\, \widehat{v}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \quad \forall\, \widehat{v} \in H_0^1(D_{\mathrm{ref}})$$

transported coefficient　　　　　　　　transported source term

$$u(\cdot,\boldsymbol{y}) = \widehat{u}(V^{-1}(\cdot,\boldsymbol{y}),\boldsymbol{y}) \iff \widehat{u}(\cdot,\boldsymbol{y}) = u(V(\cdot,\boldsymbol{y}),\boldsymbol{y})$$

Harbrecht, Peters, Siebenmorgen (2016),

Hakula, Harbrecht, Kaarnioja, Kuo, Sloan (Numer. Maths. 2024), . . .

Modeling random fields

- **Affine**

$$a(\boldsymbol{x},\boldsymbol{y}) = \psi_0(\boldsymbol{x}) + \sum_{j\geq 1} y_j\, \psi_j(\boldsymbol{x})$$

Cohen, DeVore, Schwab (2010)

$y_j \in [-\frac{1}{2}, \frac{1}{2}]$ uniform

- **Lognormal** (Karhunen–Loève expansion, circulant embedding, H-matrix)

$$a(\boldsymbol{x},\boldsymbol{y}) = \exp\Big(\psi_0(\boldsymbol{x}) + \sum_{j\geq 1} y_j\, \psi_j(\boldsymbol{x})\Big)$$

$y_j \in \mathbb{R}$ normal

Graham, Kuo, Nuyens, Scheichl, Sloan (2011, 2018)
Graham, Kuo, Nichols, Scheichl, Schwab, Sloan (2015)
Feischl, Kuo, Sloan (2018)

Modeling random fields

- **Affine**

$$a(\boldsymbol{x},\boldsymbol{y}) = \psi_0(\boldsymbol{x}) + \sum_{j\geq 1} y_j\, \psi_j(\boldsymbol{x})$$

Cohen, DeVore, Schwab (2010)

$y_j \in [-\frac{1}{2}, \frac{1}{2}]$ uniform

or

$y_j \in [-1, 1]$ Chebyshev

Adcock, Brugiapaglia, Webster (2022)

- **Lognormal** (Karhunen–Loève expansion, circulant embedding, H-matrix)

$$a(\boldsymbol{x},\boldsymbol{y}) = \exp\left(\psi_0(\boldsymbol{x}) + \sum_{j\geq 1} y_j\, \psi_j(\boldsymbol{x})\right)$$

$y_j \in \mathbb{R}$ normal

Graham, Kuo, Nuyens, Scheichl, Sloan (2011, 2018)
Graham, Kuo, Nichols, Scheichl, Schwab, Sloan (2015)
Feischl, Kuo, Sloan (2018)

- **Periodic**

$$a(\boldsymbol{x},\boldsymbol{y}) = \psi_0(\boldsymbol{x}) + \sum_{j\geq 1} \sin(2\pi y_j)\, \psi_j(\boldsymbol{x})$$

$y_j \in [0, 1]$ uniform

Kaarnioja, Kuo, Sloan (2020)

Key features of parametric PDE problems

- PDE solution $\boldsymbol{u(x,y)}$ is infinitely smooth w.r.t. $\boldsymbol{y}$ ("differentiate" the weak form)
- Truncate to $\boldsymbol{s}$ terms and solve by finite element method $\implies$ $\boldsymbol{u_{s,h}(x,y)}$
- Interested in
  - **Integration** — expected value of a linear functional $\mathbb{E}_{\boldsymbol{y}}[\mathcal{G}(u_{s,h}(\cdot,\boldsymbol{y}))]$
  - **Function approximation** — solution $u_{s,h}(\boldsymbol{x}^\dagger,\boldsymbol{y})$ as a function of $\boldsymbol{y}$
- Assume "$\boldsymbol{p}$-summability": $\boxed{\sum_{j\ge 1}\|\psi_j\|_\infty^{\boldsymbol{p}} < \infty,\ \boldsymbol{p}\in(0,1)}$

  Small $\boldsymbol{p}$ $\implies$ faster decay $\implies$ low effective dimension $\implies$ higher order convergence
- Freedom to choose the QMC function space setting for error analysis
  - The function space is not given
  - The smoothness parameter is not given
  - The function space weights are not given
  - The QMC rules are not given
- Goal: best convergence rate with constant independent of dimension $\boldsymbol{s}$

“The” elliptic PDE example (regularity of the PDE solution)

- Elliptic PDE

$$-\nabla \cdot (a(\boldsymbol{x},\boldsymbol{y})\,\nabla u(\boldsymbol{x},\boldsymbol{y})) = f(\boldsymbol{x}) \qquad \boldsymbol{x} \in D \subset \mathbb{R}^d,\ d = 1,2,3$$
$$u(\boldsymbol{x},\boldsymbol{y}) = 0 \qquad \boldsymbol{x} \in \partial D$$

(a) Affine uniform random field model Cohen, DeVore, Schwab (2010)

$$\|\partial^{\boldsymbol{\nu}} u(\cdot,\boldsymbol{y})\|_{H_0^1(D)} \le \frac{\|f\|_{H^{-1}(D)}}{a_{\min}}\,|\boldsymbol{\nu}|!\,\prod_{j\ge 1} b_j^{\nu_j}, \qquad b_j = \frac{\|\psi_j\|_\infty}{a_{\min}}$$

(b) Periodic random field model Kaarnioja, Kuo, Sloan (2020)

$$\|\partial^{\boldsymbol{\nu}} u(\cdot,\boldsymbol{y})\|_{H_0^1(D)} \le \frac{\|f\|_{H^{-1}(D)}}{a_{\min}}\,(2\pi)^{|\boldsymbol{\nu}|} \sum_{\boldsymbol{m}\le\boldsymbol{\nu}} |\boldsymbol{m}|! \prod_{j\ge 1} \left(b_j^{m_j}\,\mathcal{S}(\nu_j,m_j)\right)$$

Stirling numbers of the second kind

"The" elliptic PDE example (error analysis)

$$\int_{(-\frac{1}{2},\frac{1}{2})^{\mathbb{N}}} \mathcal{G}(u(\cdot,\boldsymbol{y}))\,\mathrm{d}\boldsymbol{y} \approx \frac{1}{N}\sum_{k=0}^{N-1} \mathcal{G}(u_{s,h}(\cdot,\boldsymbol{t}_k - \tfrac{\boldsymbol{1}}{\boldsymbol{2}}))$$

(1) dimension truncation

(2) FE discretization

(3) QMC quadrature

$$\boxed{\text{(rms) error} \le \underbrace{\text{truncation error}}_{s^{-2/p+1}} + \underbrace{\text{FE error}}_{h^2} + \underbrace{\text{(rms) QMC error}}_{n^{-r}}}$$

$$r = \begin{cases} \min(\frac{1}{p}-\frac{1}{2}, 1-\delta) \\ \frac{1}{p} \\ \frac{1}{p} \end{cases}$$

affine uniform random field + "QMC Setting 1"
Kuo, Schwab, Sloan (2012)

affine uniform random field + "QMC Setting 3"
Dick, Kuo, Le Gia, Nuyens, Schwab, Sloan (2014)

periodic random field + "QMC Setting 4"
Kaarnioja, Kuo, Sloan (2020)

- Key assumptions:
  - $\sum_{j\ge1} \|\psi_j\|_\infty^p < \infty$ for some $p \in (0,1)$
  - first order FEM; $f, \mathcal{G} \in L_2(D)$
- (rms) **QMC error** $\le$ (rms) **worst case error**$_\gamma$ $\times$ $\|\mathcal{G}(u_{s,h})\|_\gamma$

Lattice Rules

Kernel Methods

(SIDE STEP: Parametric PDEs in UQ)

Deep Neural Networks

Data-to-Observables map

- Given data $\boldsymbol{y} \in \boldsymbol{Y} \subset \mathbb{R}^{s}$, compute observable $G(\boldsymbol{y}) \in \mathbb{R}^{N_{\mathrm{obs}}}$
  - A vector of s input parameters to a parametric PDE
  - A vector of N_{obs} observables obtained from the solution to a PDE

    Examples
    - Average of the PDE solution over a subset of the domain $\implies N_{\mathrm{obs}} = 1$
    - Vector of PDE solutions at finite element mesh $\implies N_{\mathrm{obs}} = \#\mathrm{FE\ nodes}$

- **Deep Neural Network** (DNN) as function approximation
  - $G(\boldsymbol{y}) \approx G_\theta^{[L]}(\boldsymbol{y})$
  - Want to have a small $\|G - G_\theta^{[L]}\|_{L_2}$

Deep Neural Network (DNN)

Standard "feed-forward" DNN

$$G_{\boldsymbol{\theta}}^{[L]}(\boldsymbol{y}) := W_L\,\sigma(\cdots W_2\,\sigma(W_1\,\sigma(W_0\,\boldsymbol{y} + \boldsymbol{v}_0) + \boldsymbol{v}_1) + \boldsymbol{v}_2 \cdots) + \boldsymbol{v}_L$$

- Input: $\boldsymbol{y} \in Y := [0,1]^s$, $d_0 = s$ is the dimension of the input vector $\boldsymbol{y}$
- Output: $G_{\boldsymbol{\theta}}^{[L]}(\boldsymbol{y}) \in \mathbb{R}^{N_{\rm obs}}$, $d_{L+1} = N_{\rm obs}$ is the number of observables
- Unknowns to be "learned": $\boldsymbol{\theta} := \{(W_\ell, \boldsymbol{v}_\ell)\}_{\ell=0}^{L} \in \Theta$ "layer" ℓ
  - W_ℓ is a $d_{\ell+1} \times d_\ell$ "weight" matrix
  - $\boldsymbol{v}_\ell$ is a $d_{\ell+1} \times 1$ "bias" vector
- Hyperparameters (network architecture): "depth" L, "width" d_ℓ, ...
- σ is a univariate activation function: $\sigma(x) = \max(x,0)$ (rectified linear unit / ReLU), $\sigma(x) = \frac{1}{1+e^{-x}}$ (logistic sigmoid), $\sigma(x) = \tanh(x)$, $\sigma(x) = \frac{x}{1+e^{-x}}$ (swish)

Deep Neural Network (DNN)

(a) Non-periodic DNN

$$G_{\boldsymbol{\theta}}^{[L]}(\boldsymbol{y}) := W_L\, \sigma(\cdots W_2\, \sigma(W_1\, \sigma(W_0\, \boldsymbol{y} + \boldsymbol{v}_0) + \boldsymbol{v}_1) + \boldsymbol{v}_2 \cdots) + \boldsymbol{v}_L$$

(b) Periodic DNN

Keller, Kuo, Nuyens, Sloan (submitted 2025)

$$G_{\boldsymbol{\theta}}^{[L]}(\boldsymbol{y}) := W_L\, \sigma(\cdots W_2\, \sigma(W_1\, \sigma(W_0\, \sin(2\pi \boldsymbol{y}) + \boldsymbol{v}_0) + \boldsymbol{v}_1) + \boldsymbol{v}_2 \cdots) + \boldsymbol{v}_L$$

- Input: $\boldsymbol{y} \in Y := [0,1]^s$, $d_0 = s$ is the dimension of the input vector $\boldsymbol{y}$
- Output: $G_{\boldsymbol{\theta}}^{[L]}(\boldsymbol{y}) \in \mathbb{R}^{N_{\rm obs}}$, $d_{L+1} = N_{\rm obs}$ is the number of observables
- Unknowns to be “learned”: $\boldsymbol{\theta} := \{(W_\ell, \boldsymbol{v}_\ell)\}_{\ell=0}^{L} \in \Theta$ “layer” ℓ
  - W_ℓ is a $d_{\ell+1} \times d_\ell$ “weight” matrix
  - $\boldsymbol{v}_\ell$ is a $d_{\ell+1} \times 1$ “bias” vector
- Hyperparameters (network architecture): “depth” L, “width” d_ℓ, ...
- σ is a univariate activation function: $\sigma(x) = \max(x, 0)$ (rectified linear unit / ReLU), $\sigma(x) = \frac{1}{1+e^{-x}}$ (logistic sigmoid), $\sigma(x) = \tanh(x)$, $\sigma(x) = \frac{x}{1+e^{-x}}$ (swish)

Error analysis for DNN

Mishra, Rusch (2021)
Longo, Mishra, Rusch, Schwab (2021)

- "Training" – to "learn" network parameters

$$\boldsymbol{\theta}^* := \operatorname{argmin}_{\boldsymbol{\theta}\in\Theta}\Big(\mathcal{J}(\boldsymbol{\theta}) + \lambda\,\mathcal{R}(\boldsymbol{\theta})\Big), \quad \mathcal{J}(\boldsymbol{\theta}) := \frac{1}{N}\sum_{k=0}^{N-1} |G(\boldsymbol{t}_k) - G_{\boldsymbol{\theta}}^{[L]}(\boldsymbol{t}_k)|^2$$

Training points

- **Generalization error**

$$\mathcal{E}_G := \left(\int_{[0,1]^s} |G(\boldsymbol{y}) - G_{\boldsymbol{\theta}}^{[L]}(\boldsymbol{y})|^2\,\mathrm{d}\boldsymbol{y}\right)^{1/2} = \|G - G_{\boldsymbol{\theta}}^{[L]}\|_{L_2}$$

- **Training error**

$$\mathcal{E}_T := \left(\frac{1}{N}\sum_{k=0}^{N-1} |G(\boldsymbol{t}_k) - G_{\boldsymbol{\theta}}^{[L]}(\boldsymbol{t}_k)|^2\right)^{1/2}$$

Error analysis for DNN

Mishra, Rusch (2021)
Longo, Mishra, Rusch, Schwab (2021)

- "Training" – to "learn" network parameters

$$\theta^* := \mathrm{argmin}_{\theta\in\Theta}\Big(\mathcal{J}(\theta) + \lambda\,\mathcal{R}(\theta)\Big), \quad \mathcal{J}(\theta) := \frac{1}{N}\sum_{k=0}^{N-1} |G(\boldsymbol{t}_k) - G_\theta^{[L]}(\boldsymbol{t}_k)|^2$$

Training points

- **Generalization error**

$$\mathcal{E}_G := \bigg(\int_{[0,1]^s} |G(\boldsymbol{y}) - G_\theta^{[L]}(\boldsymbol{y})|^2\,\mathrm{d}\boldsymbol{y}\bigg)^{1/2} = \|G - G_\theta^{[L]}\|_{L_2}$$

- **Training error**

$$\mathcal{E}_T := \bigg(\frac{1}{N}\sum_{k=0}^{N-1} |G(\boldsymbol{t}_k) - G_\theta^{[L]}(\boldsymbol{t}_k)|^2\bigg)^{1/2}$$

- **Generalization gap**

Quadrature error for $F(\boldsymbol{y}) := (G(\boldsymbol{y}) - G_\theta^{[L]}(\boldsymbol{y}))^2$

$$|\mathcal{E}_G - \mathcal{E}_T| \le \sqrt{|\mathcal{E}_G^2 - \mathcal{E}_T^2|} = \bigg|\int_{[0,1]^s} (G(\boldsymbol{y}) - G_\theta^{[L]}(\boldsymbol{y}))^2\,\mathrm{d}\boldsymbol{y} - \frac{1}{N}\sum_{k=0}^{N-1}(G(\boldsymbol{t}_k) - G_\theta^{[L]}(\boldsymbol{t}_k))^2\bigg|^{\frac{1}{2}}$$

Error analysis for DNN

Mishra, Rusch (2021)
Longo, Mishra, Rusch, Schwab (2021)

- “Training” – to “learn” network parameters

$$\theta^* := \mathrm{argmin}_{\theta\in\Theta}\Big(\mathcal{J}(\theta) + \lambda\,\mathcal{R}(\theta)\Big), \quad \mathcal{J}(\theta) := \frac{1}{N}\sum_{k=0}^{N-1} |G(\boldsymbol{t}_k) - G_\theta^{[L]}(\boldsymbol{t}_k)|^2$$

Training points

- **Generalization error**

$$\mathcal{E}_G := \left(\int_{[0,1]^s} |G(\boldsymbol{y}) - G_\theta^{[L]}(\boldsymbol{y})|^2\,\mathrm{d}\boldsymbol{y}\right)^{1/2} = \|G - G_\theta^{[L]}\|_{L_2}$$

- **Training error**

$$\mathcal{E}_T := \left(\frac{1}{N}\sum_{k=0}^{N-1} |G(\boldsymbol{t}_k) - G_\theta^{[L]}(\boldsymbol{t}_k)|^2\right)^{1/2}$$

- **Generalization gap**

Quadrature error for $F(\boldsymbol{y}) := (G(\boldsymbol{y}) - G_\theta^{[L]}(\boldsymbol{y}))^2$

$$|\mathcal{E}_G - \mathcal{E}_T| \le \sqrt{|\mathcal{E}_G^2 - \mathcal{E}_T^2|} = \left|\int_{[0,1]^s} (G(\boldsymbol{y}) - G_\theta^{[L]}(\boldsymbol{y}))^2\,\mathrm{d}\boldsymbol{y} - \frac{1}{N}\sum_{k=0}^{N-1}(G(\boldsymbol{t}_k) - G_\theta^{[L]}(\boldsymbol{t}_k))^2\right|^{\frac{1}{2}}$$

- Theoretical bound for generalization error

Need to know the regularity of $(G_\theta^{[L]}(\boldsymbol{y}))^2$

$$\boxed{\mathcal{E}_G \le \mathcal{E}_T + |\mathcal{E}_G - \mathcal{E}_T| \le \mathcal{E}_T + \left(e^{\mathrm{wor-int}} \times \|(G - G_\theta^{[L]})^2\|_{\mathcal{W}_{\alpha,\boldsymbol{\gamma}}}\right)^{1/2}}$$

Key assumptions

Keller, Kuo, Nuyens, Sloan (submitted 2025)

(a) Non-periodic DNN

$$\begin{cases} G_{\theta}^{[0]}(\boldsymbol{y}) := \boldsymbol{W}_0\, \boldsymbol{y} + \boldsymbol{v}_0 \\ G_{\theta}^{[\ell]}(\boldsymbol{y}) := \boldsymbol{W}_\ell\, \boldsymbol{\sigma}(G_{\theta}^{[\ell-1]}(\boldsymbol{y})) + \boldsymbol{v}_\ell \quad \text{for} \quad \ell \geq 1 \end{cases}$$

(b) Periodic DNN

$$\begin{cases} G_{\theta}^{[0]}(\boldsymbol{y}) := \boldsymbol{W}_0\, \sin(2\pi \boldsymbol{y}) + \boldsymbol{v}_0 \\ G_{\theta}^{[\ell]}(\boldsymbol{y}) := \boldsymbol{W}_\ell\, \boldsymbol{\sigma}(G_{\theta}^{[\ell-1]}(\boldsymbol{y})) + \boldsymbol{v}_\ell \quad \text{for} \quad \ell \geq 1 \end{cases}$$

- Assumption 1 The columns of the matrix $\boldsymbol{W}_0$ have bounded vector ∞-norms
  $\|\boldsymbol{W}_{0,:,j}\|_\infty \leq \beta_j$ for all $j \in \{1 : s\}$

  Cf. Longo, Mishra, Rusch, Schwab (2021)

- Assumption 2 The matrices $\boldsymbol{W}_\ell$ for $\ell \geq 1$ have bounded matrix ∞-norms
  $\|\boldsymbol{W}_\ell\|_\infty \leq R_\ell$ for all $\ell \geq 1$

  Cf. Longo, Mishra, Rusch, Schwab (2021)

- Assumption 3 The activation function $\boldsymbol{\sigma}$ is smooth and satisfies
  $\|\boldsymbol{\sigma}^{(n)}\|_\infty \leq A_n$ for all $n \geq 1$

Regularity bounds for DNNs

Theorem 1 (Keller, Kuo, Nuyens, Sloan) For any multiindex $\boldsymbol{\nu} \neq \mathbf{0}$

(a) Non-periodic DNN

$$|\partial^{\boldsymbol{\nu}} G_\theta^{[L]}(\boldsymbol{y})| \leq R_L\, \Gamma_{|\boldsymbol{\nu}|}^{[L]} \prod_{j=1}^{s} \beta_j^{\nu_j}$$

(b) Periodic DNN

$$|\partial^{\boldsymbol{\nu}} G_\theta^{[L]}(\boldsymbol{y})| \leq R_L \sum_{\boldsymbol{m} \leq \boldsymbol{\nu}} \Gamma_{|\boldsymbol{m}|}^{[L]} \prod_{j=1}^{s} \left(\beta_j^{m_j} \mathcal{S}(\nu_j, m_j)\right)$$

Stirling numbers of the second kind

- The sequence $\Gamma_n^{[\ell]}$ is defined recursively by
  - $\Gamma_n^{[1]} := A_n$ for $n \geq 1$, $\Gamma_n^{[\ell]} := \sum_{\lambda=1}^{n} A_\lambda\, R_{\ell-1}^{\lambda}\, \mathbb{B}_{n,\lambda}^{[\ell-1]}$ for $\ell \geq 2$ and $n \geq 1$
  - $\mathbb{B}_{n,1}^{[\ell]} := \Gamma_n^{[\ell]}$ for $n \geq 1$, $\mathbb{B}_{n,\lambda}^{[\ell]} := \sum_{i=\lambda-1}^{n-1} \binom{n-1}{i} \Gamma_{n-i}^{[\ell]}\, \mathbb{B}_{i,\lambda-1}^{[\ell]}$ for $\ell \geq 1$ and $n \geq \lambda \geq 2$

Regularity bounds for DNNs

Theorem 2 (Keller, Kuo, Nuyens, Sloan) For any multiindex $\boldsymbol{\nu}$ and $A_n = \xi\,\tau^n\,n!$

(a) Non-periodic DNN

$$|\partial^{\boldsymbol{\nu}} G_\theta^{[L]}(\boldsymbol{y})| \le C_L\,|\boldsymbol{\nu}|! \prod_{j=1}^{s} (S_L\beta_j)^{\nu_j}, \qquad S_L := \tau \sum_{\ell=0}^{L-1} \prod_{k=1}^{\ell} (\xi\,\tau R_k)$$

(b) Periodic DNN

$$|\partial^{\boldsymbol{\nu}} G_\theta^{[L]}(\boldsymbol{y})| \le C_L \sum_{\boldsymbol{m}\le\boldsymbol{\nu}} |\boldsymbol{m}|! \prod_{j=1}^{s} \Big((S_L\beta_j)^{m_j}\,\mathcal{S}(\nu_j, m_j)\Big)$$

- Common activation functions

$$\begin{cases} \text{sigmoid:} & \sigma(x) = \frac{1}{1+e^{-x}} & \sigma^{(n)}(x) \le A_n = n! & (\xi = 1,\ \tau = 1) \\ \text{tanh:} & \sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} & \sigma^{(n)}(x) \le A_n = 2^n\,n! & (\xi = 1,\ \tau = 2) \\ \text{swish:} & \sigma(x) = \frac{x}{1+e^{-x}} & \sigma^{(n)}(x) \le A_n = 1.1\,n! & (\xi = 1.1,\ \tau = 1) \\ \text{swish}_c\text{:} & \sigma(x) = \frac{x}{1+e^{-c\,x}} & \sigma^{(n)}(x) \le A_n = \frac{1.1}{c}\,c^n\,n! & (\xi = \frac{1.1}{c},\ \tau = c) \end{cases}$$

- $S_L = L$ for sigmoid, when $R_\ell \le 1$ for all $\ell \in \{1 : L-1\}$

Norm bounds for DNNs

Theorem 3 (Keller, Kuo, Nuyens, Sloan)

(a) Non-periodic DNN

$$\|(G_\theta^{[L]})^2\|^2_{\mathcal{W}_{1,\boldsymbol{\gamma}}} \le C_L^4 \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{1}{\gamma_\mathfrak{u}} \bigg((|\mathfrak{u}|+1)! \prod_{j\in\mathfrak{u}} (S_L \beta_j) \bigg)^2$$

(b) Periodic DNN

$$\|(G_\theta^{[L]})^2\|^2_{\mathcal{W}_{\alpha,\boldsymbol{\gamma}}} \le C_L^4 \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{(2\pi)^{2\alpha|\mathfrak{u}|}}{\gamma_\mathfrak{u}} \bigg(\sum_{\boldsymbol{m}_\mathfrak{u}\le\boldsymbol{\alpha}_\mathfrak{u}} (|\boldsymbol{m}_\mathfrak{u}|+1)! \prod_{j\in\mathfrak{u}} \big((S_L\beta_j)^{m_j} \mathcal{S}(\alpha, m_j)\big) \bigg)^2$$

Norm bounds for DNNs

Theorem 3 (Keller, Kuo, Nuyens, Sloan)

(a) **Non-periodic DNN + PDE with affine uniform random field**

$$\|(G_\theta^{[L]})^2\|_{\mathcal{W}_{1,\gamma}}^2 \le C_L^4 \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{1}{\gamma_\mathfrak{u}} \bigg((|\mathfrak{u}|+1)! \prod_{j\in\mathfrak{u}} (S_L\beta_j)\bigg)^2$$

$$\|G^2\|_{\mathcal{W}_{1,\gamma}}^2 \le C^4 \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{1}{\gamma_\mathfrak{u}} \bigg((|\mathfrak{u}|+1)! \prod_{j\in\mathfrak{u}} b_j\bigg)^2, \quad b_j := \frac{\|\psi_j\|_\infty}{a_{\min}}$$

(b) **Periodic DNN + PDE with periodic random field**

$$\|(G_\theta^{[L]})^2\|_{\mathcal{W}_{\alpha,\gamma}}^2 \le C_L^4 \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{(2\pi)^{2\alpha|\mathfrak{u}|}}{\gamma_\mathfrak{u}} \bigg(\sum_{\boldsymbol{m}_\mathfrak{u}\le\boldsymbol{\alpha}_\mathfrak{u}} (|\boldsymbol{m}_\mathfrak{u}|+1)! \prod_{j\in\mathfrak{u}} \big((S_L\beta_j)^{m_j}\mathcal{S}(\alpha,m_j)\big)\bigg)^2$$

$$\|G^2\|_{\mathcal{W}_{\alpha,\gamma}}^2 \le C^4 \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{(2\pi)^{2\alpha|\mathfrak{u}|}}{\gamma_\mathfrak{u}} \bigg(\sum_{\boldsymbol{m}_\mathfrak{u}\le\boldsymbol{\alpha}_\mathfrak{u}} (|\boldsymbol{m}_\mathfrak{u}|+1)! \prod_{j\in\mathfrak{u}} \big(b_j^{m_j}\mathcal{S}(\alpha,m_j)\big)\bigg)^2$$

Norm bounds for the generalization gap

Theorem 4 (Keller, Kuo, Nuyens, Sloan)

- restrict the elements of $\boldsymbol{W}_\ell$ so that $\boldsymbol{R}_\ell \le \boldsymbol{\rho}$ for all $\boldsymbol{\ell} \in \{1:L-1\}$
- restrict the elements of $\boldsymbol{W}_0$ so that $\boldsymbol{S}_L\boldsymbol{\beta}_j \le \boldsymbol{b}_j := \dfrac{\|\psi_j\|_\infty}{a_{\min}}$ for all $\boldsymbol{j} \in \{1:s\}$
- restrict the elements of $\boldsymbol{W}_L$ and $\boldsymbol{v}_L$ so that $\boldsymbol{C}_L \le \boldsymbol{C}$

(a) Non-periodic DNN + PDE with affine uniform random field

$$\|(G - G_\theta^{[L]})^2\|^2_{\mathcal{W}_{1,\boldsymbol{\gamma}}} \le 2\,C^4 \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{1}{\gamma_\mathfrak{u}} \Big((|\mathfrak{u}|+1)! \prod_{j\in\mathfrak{u}} b_j\Big)^2$$

(b) Periodic DNN + PDE with periodic random field

$$\|(G - G_\theta^{[L]})^2\|^2_{\mathcal{W}_{\alpha,\boldsymbol{\gamma}}} \le 2\,C^4 \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{(2\pi)^{2\alpha|\mathfrak{u}|}}{\gamma_\mathfrak{u}} \Big(\sum_{\boldsymbol{m}_\mathfrak{u}\le\boldsymbol{\alpha}_\mathfrak{u}} (|\boldsymbol{m}_\mathfrak{u}|+1)! \prod_{j\in\mathfrak{u}} (b_j^{m_j}\,\mathcal{S}(\alpha,m_j))\Big)^2$$

Norm bounds for the generalization gap

Theorem 4 (Keller, Kuo, Nuyens, Sloan)

- restrict the elements of W_ℓ so that $R_\ell \le \rho$ for all $\ell \in \{1 : L-1\}$
- restrict the elements of W_0 so that $S_L \beta_j \le b_j := \dfrac{\|\psi_j\|_\infty}{a_{\min}}$ for all $j \in \{1 : s\}$
- restrict the elements of W_L and $\boldsymbol{v}_L$ so that $C_L \le C$

(a) Non-periodic DNN + PDE with affine uniform random field

$$\|(G - G_\theta^{[L]})^2\|_{\mathcal{W}_{1,\boldsymbol{\gamma}}}^2 \le 2\,C^4 \sum_{\mathfrak{u} \subseteq \{1:s\}} \frac{1}{\gamma_{\mathfrak{u}}} \bigg((|\mathfrak{u}|+1)! \prod_{j \in \mathfrak{u}} b_j \bigg)^2 \qquad \boxed{e^{\mathrm{wor-int}} = \mathcal{O}(N^{-1+\delta})}$$

(b) Periodic DNN + PDE with periodic random field

$$\|(G - G_\theta^{[L]})^2\|_{\mathcal{W}_{\alpha,\boldsymbol{\gamma}}}^2 \le 2\,C^4 \sum_{\mathfrak{u} \subseteq \{1:s\}} \frac{(2\pi)^{2\alpha|\mathfrak{u}|}}{\gamma_{\mathfrak{u}}} \bigg(\sum_{\boldsymbol{m}_{\mathfrak{u}} \le \boldsymbol{\alpha}_{\mathfrak{u}}} (|\boldsymbol{m}_{\mathfrak{u}}|+1)! \prod_{j \in \mathfrak{u}} \big(b_j^{m_j}\, \mathcal{S}(\alpha, m_j)\big) \bigg)^2$$

$$\boxed{e^{\mathrm{wor-int}} = \mathcal{O}(N^{-\alpha+\delta})}$$

- Theoretical bound for generalization error

$$\boxed{\mathcal{E}_G \le \mathcal{E}_T + |\mathcal{E}_G - \mathcal{E}_T| \le \mathcal{E}_T + \Big(e^{\mathrm{wor-int}} \times \|(G - G_\theta^{[L]})^2\|_{\mathcal{W}_{\alpha,\boldsymbol{\gamma}}} \Big)^{1/2} \le \mathtt{tol} + \mathcal{O}(N^{-r/2})}$$

Numerical experiments

- Algebraic equation $a(\boldsymbol{y})\,G(\boldsymbol{y}) = 1$ mimicking parametric PDE, “decay” $q = 2.5$,

$$G(\boldsymbol{y}) = \frac{1}{a(\boldsymbol{y})}, \quad a(\boldsymbol{y}) = 1 + \sum_{j=1}^{s} \sin(2\pi y_j)\,\psi_j, \quad \psi_j = \frac{0.5}{j^q}, \quad b_j = \frac{0.5}{(1-0.5\,\zeta(q))\,j^q}$$

- Hyperparameters: 1. $L = 3$, $N_{\rm obs} = 1$, $d_\ell = 32$, $s = 50$ (3777 parameters)
  2. $L = 12$, $N_{\rm obs} = 1$, $d_\ell = 30$, $s = 50$ (11791 parameters)

- **Tailored regularization** to “encourage” $\beta_j \le \dfrac{b_j}{L}$ for all $j \in \{1:s\}$

$$\theta^* = \mathrm{argmin}_{\theta\in\Theta}\Big(\tfrac{1}{N}\textstyle\sum_{k=0}^{N-1}\big(G(\boldsymbol{t}_k) - G_\theta^{[L]}(\boldsymbol{t}_k)\big)^2 + \lambda\,\|\theta\|_2^2 + \lambda_1\,\mathcal{R}_1(\theta)\Big)$$

$$\mathcal{R}_1(\theta) = \frac{1}{s}\sum_{j=1}^{s}\frac{1}{d_1}\sum_{p=1}^{d_1}\left(W_{0,p,j}^2\,\frac{L^2}{b_j^2}\right)^{m/2}, \quad m = 6$$

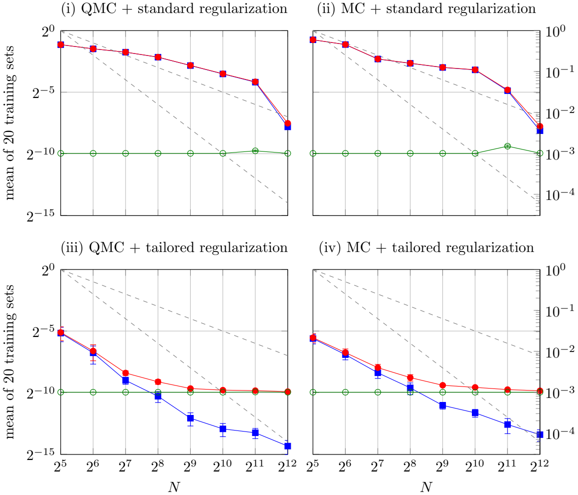
  - PyTorch. Full-batch ADAM. Random Glorot initialization.
  - Learning rate (step size) 10^{-4}. Regularization parameters $\lambda = 10^{-8}$, $\lambda_1 = 10^{-8}$.
  - Stop when training error reaches $\texttt{tol} = 10^{-3}$ or maximum 40000 epochs (steps).
  - Off-the-shelf embedded lattice point set with $N = 2^5, \ldots, 2^{12}$ points.
  - Different point set with $M = 2^{15}$ points to estimate the generalization error.

1. $L = 3$, $N_{\mathrm{obs}} = 1$, $d_\ell = 32$, $s = 50$ (3777 parameters)

$$\widetilde{\mathcal{E}}_G \le \mathcal{E}_T + |\widetilde{\mathcal{E}}_G - \mathcal{E}_T| \le \texttt{tol} + \mathcal{O}(N^{-r/2})$$

$\mathcal{E}_T$ training error
$\widetilde{\mathcal{E}}_G$ (estimated) generalization error
$|\widetilde{\mathcal{E}}_G - \mathcal{E}_T|$ (estimated) generalization gap

sigmoid activation function

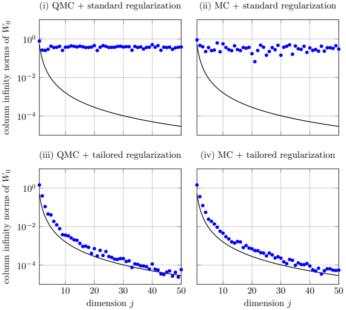


1. $L = 3$, $N_{\text{obs}} = 1$, $d_\ell = 32$, $s = 50$ (3777 parameters)

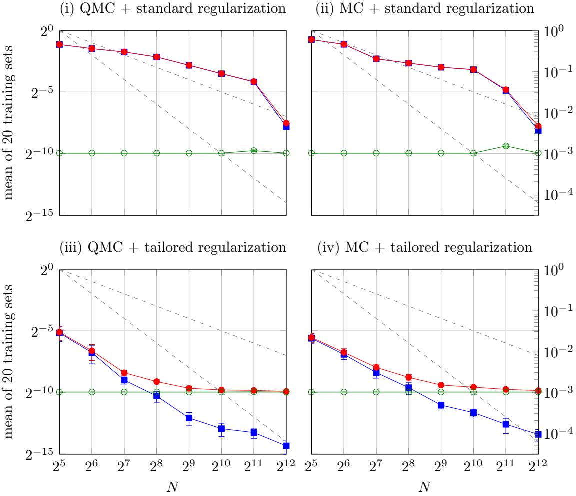
$$\boxed{\widetilde{\mathcal{E}}_G \le \mathcal{E}_T + |\widetilde{\mathcal{E}}_G - \mathcal{E}_T| \le \mathtt{tol} + \mathcal{O}(N^{-r/2})}$$

$\mathcal{E}_T$ training error
$\widetilde{\mathcal{E}}_G$ (estimated) generalization error
$|\widetilde{\mathcal{E}}_G - \mathcal{E}_T|$ (estimated) generalization gap

sigmoid activation function



$\log(\beta_j)$ v.s. $\log(b_j/L)$

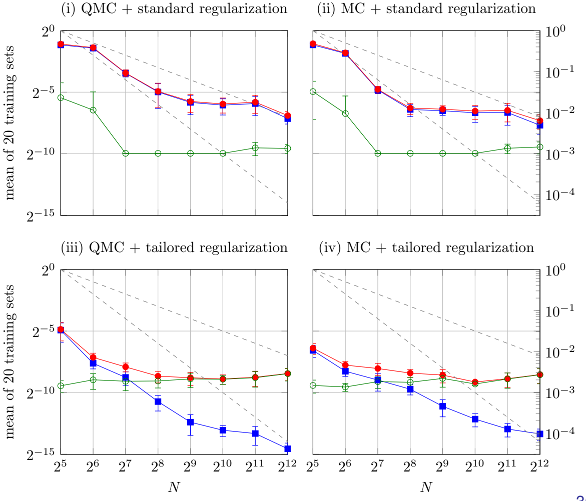


2. $L = 12$, $N_{\mathrm{obs}} = 1$, $d_\ell = 30$, $s = 50$ (11791 parameters)

$$\boxed{\widetilde{\mathcal{E}}_G \le \mathcal{E}_T + |\widetilde{\mathcal{E}}_G - \mathcal{E}_T| \le \mathtt{tol} + \mathcal{O}(N^{-r/2})}$$

$\mathcal{E}_T$ training error
$\widetilde{\mathcal{E}}_G$ (estimated) generalization error
$|\widetilde{\mathcal{E}}_G - \mathcal{E}_T|$ (estimated) generalization gap

sigmoid activation function

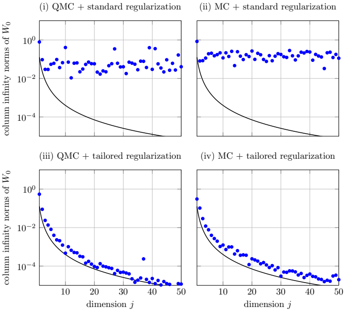


2. $L = 12$, $N_{\rm obs} = 1$, $d_\ell = 30$, $s = 50$ (11791 parameters)

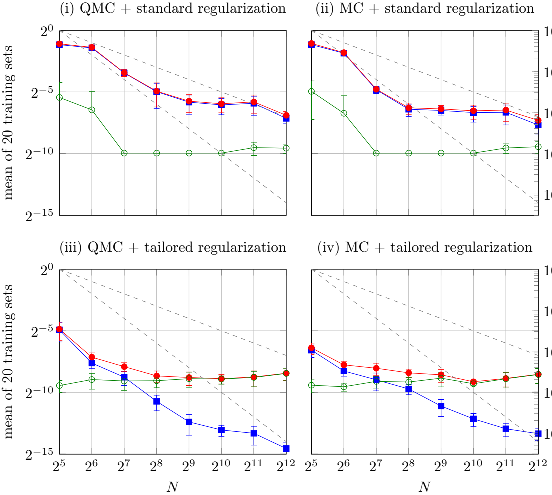
$$\boxed{\widetilde{\mathcal{E}}_G \le \mathcal{E}_T + |\widetilde{\mathcal{E}}_G - \mathcal{E}_T| \le \texttt{tol} + \mathcal{O}(N^{-r/2})}$$

$\mathcal{E}_T$ training error
$\widetilde{\mathcal{E}}_G$ (estimated) generalization error
$|\widetilde{\mathcal{E}}_G - \mathcal{E}_T|$ (estimated) generalization gap

sigmoid activation function

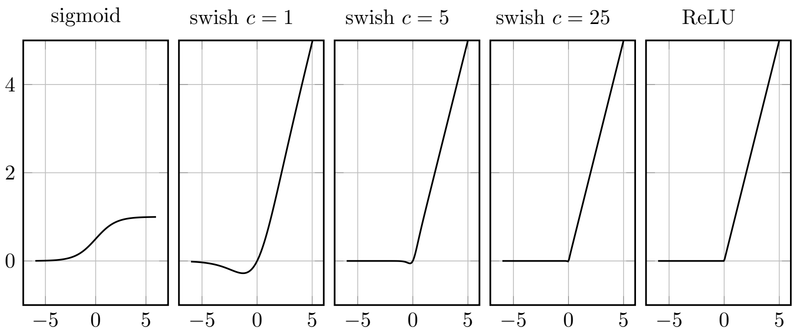


$\log(\beta_j)$ v.s. $\log(b_j/L)$



Different activation functions

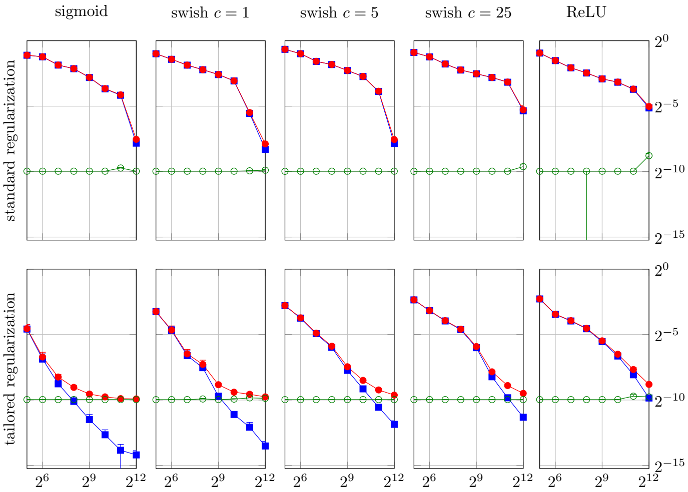
$$\textbf{sigmoid}(x) = \frac{1}{1+e^{-x}} \qquad \textbf{swish}_{c}(x) = \frac{x}{1+e^{-c\,x}} \qquad \textbf{ReLU}(x) = \max(x, 0)$$



1. $L = 3$, $N_{\rm obs} = 1$, $d_\ell = 32$, $s = 50$ (3777 parameters)

$$\boxed{\widetilde{\mathcal{E}}_G \le \mathcal{E}_T + |\widetilde{\mathcal{E}}_G - \mathcal{E}_T| \le \mathtt{tol} + \mathcal{O}(N^{-r/2})}$$

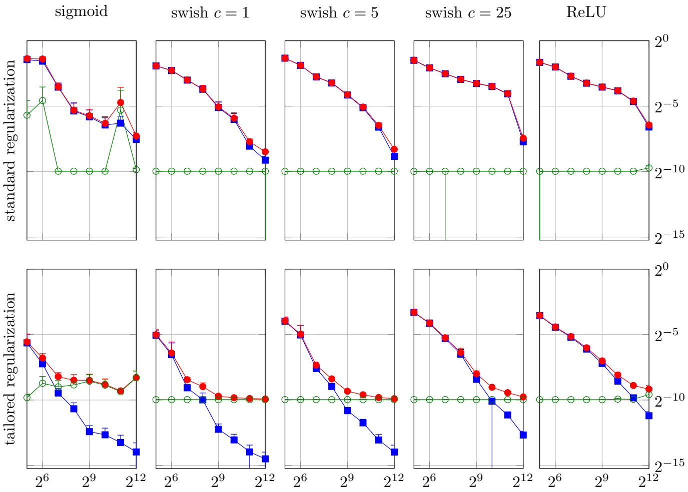
$\mathcal{E}_T$ training error
$\widetilde{\mathcal{E}}_G$ (estimated) generalization error
$|\widetilde{\mathcal{E}}_G - \mathcal{E}_T|$ (estimated) generalization gap



2. $L = 12$, $N_{\text{obs}} = 1$, $d_\ell = 30$, $s = 50$ (11791 parameters)

$$\boxed{\widetilde{\mathcal{E}}_G \le \mathcal{E}_T + |\widetilde{\mathcal{E}}_G - \mathcal{E}_T| \le \texttt{tol} + \mathcal{O}(N^{-r/2})}$$

$\mathcal{E}_T$ training error
$\widetilde{\mathcal{E}}_G$ (estimated) generalization error
$|\widetilde{\mathcal{E}}_G - \mathcal{E}_T|$ (estimated) generalization gap



Summary

- When can we use QMC?
  - High dimensional integration
  - Multivariate function approximation
  - Density estimation (for CDF and PDF, using "preintegration")
  - Training points for DNNs
  - $\cdots$
- QMC have higher convergence rates with respect to N for smooth functions
- QMC error bounds can be independent of dimension s in "weighted" spaces
- QMC points can be tailored to the applications (fast CBC construction)

Summary

- When can we use QMC?
  - High dimensional integration
  - Multivariate function approximation
  - Density estimation (for CDF and PDF, using “preintegration”)
  - Training points for DNNs
  - $\cdots$
- QMC have higher convergence rates with respect to N for smooth functions
- QMC error bounds can be independent of dimension s in “weighted” spaces
- QMC points can be tailored to the applications (fast CBC construction)
- QMC for DNNs
  - **Explicit derivative bounds** on the non-periodic and periodic DNNs
  - **Tailor-constructed QMC** rules to ensure good convergence rates, with error bound independent of dimension
  - **Tailored regularization** during training to “encourage” network parameters to match the derivative features of the target function
  - Promising numerical results on simple test function