ANITI

DEEL
DEpendable & Explainable Learning

Towards instance-dependent approximation guarantees for Lipschitz approximators, application to Scientific ML.

P. Novello, IRT Saint Exupery
C. Gauchy, CEA
M. Dalery, Laboratoire de Mathématiques de Besançon
M. Peyron, CERFACS & EVIDEN
S. Saha, Indian Statistical Institute

IVADO · IRT Saint Exupéry · CRIAQ · ANITI Artificial & Natural Intelligence Toulouse Institute · Institut intelligence et données

Challenges of SciML

Scientific Machine Learning is thriving [2] ...

- Extends traditional surrogate modeling and function approximation to larger scale problems (mesh data) [5,7].
- Encompasses new techniques like Physics informed learning, neural operators ([5,6]., this workshop) to refine the quality of the approximation and foster practitioner's trust in those models.

...but surrogate models and numerical schemes are not considered equals

- Such models are data driven and lack guarantees as seen classical numerical schemes
- Some workaround to leverage ML without affecting the guarantees:
  - ML-driven preconditioning [9], Mesh initialization [13],...

Still, the performances of next gen surrogate models can be so good as is...

...Couldn't we provide strict **approximation guarantees for SciML models**?

Finding an error bound for g

We approximated a function $f: \mathcal{X} \in \mathbb{R}^d \to \mathbb{R}$ using a neural network g and a set of learning points $\left(X_1, Y_1 = f(X_1)\right), \dots, \left(X_n, Y_n = f(X_n)\right)$

Now, **can we provide approximation guarantees after the training using g and $\left(X_1, Y_1 = f(X_1)\right), \dots, \left(X_n, Y_n = f(X_n)\right)$ only?**

By finding bounds on

$$J_g = \|f - g\|_\infty = \max_{x \in \mathcal{X}} |f(x) - g_\theta(x)|$$

In the following, we try to bound the norm $\|f - g\|_\infty$, with a bound $\bar{J}_g$.

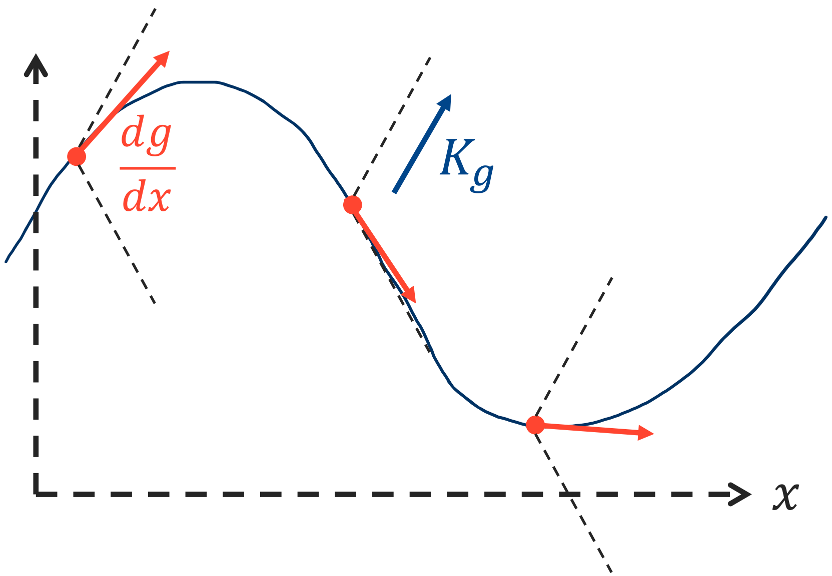
To that end, we will leverage the properties of Lipschitz neural networks

Lipschitz Neural Networks

A function f is said Lipschitz continuous, of constant K_f if :

$$\forall x, y \in \mathbb{R}^d, |f(x) - f(y)| \leq K_f \times \|x - y\|$$

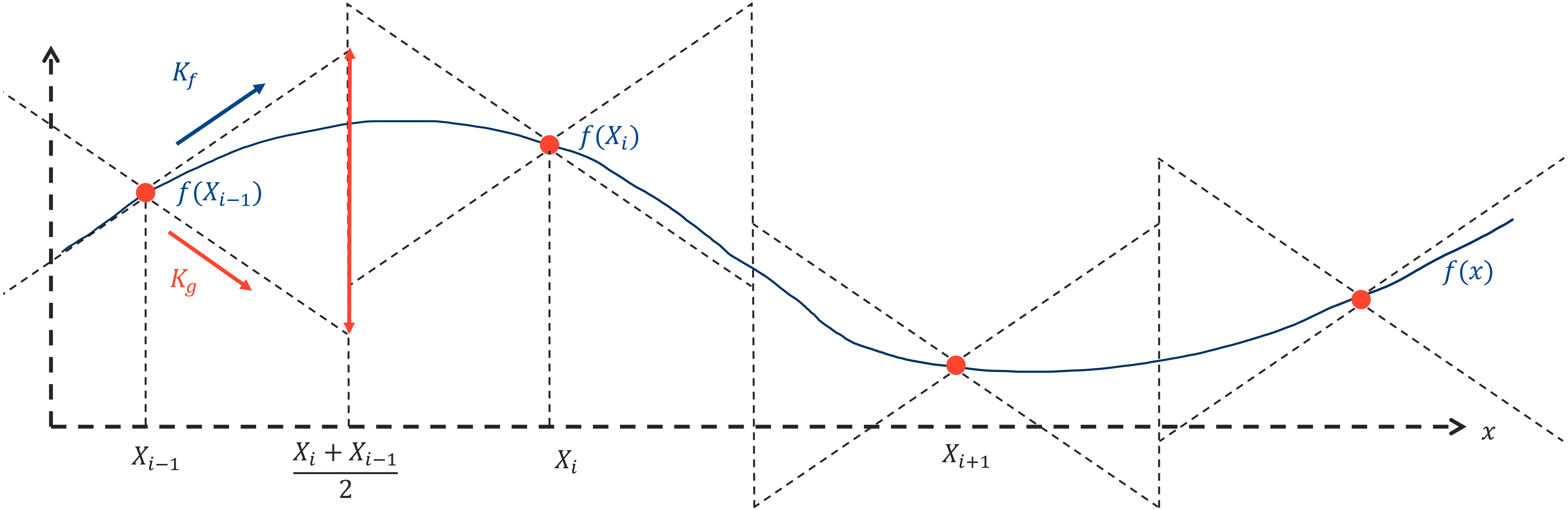
A neural network g is said K_g-Lipschitz when it satisfies the above property.



Its rate of change is bounded by K_g

Motivation: Error bound in 1D

DEEL

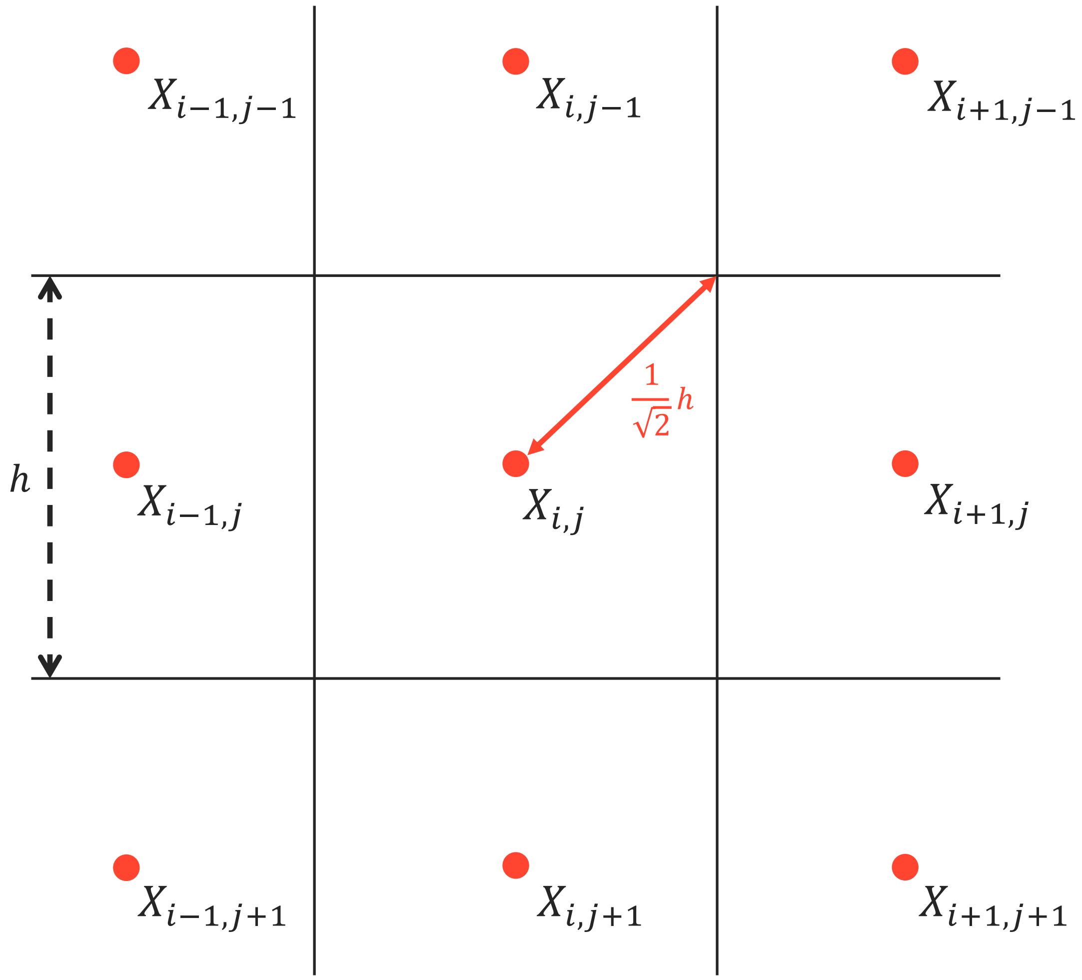


Take the difference between maximum variation of f and g on each subdivision:

$$J_g \leq \max_{i \in \{1,\dots,n\}} \frac{1}{2}\left(K_g + K_f\right)\|X_i - X_{i-1}\| + \boxed{|f(X_i) - g(X_i)|}$$

$=$ 0 in this example

Motivation: Error bound in 2D (and beyond)



Bound in 2D ($d = 2$):

- Consider n^2 learning points $\{X_{i,j}\}_{i,j\in\{1,\ldots n\}^\wedge 2}$ at the center of a grid with cells of edge size h.

In the k-th cell of center $X_{i,j}$:

$$J_g^k \leq |f(X_{i,j}) - g(X_{i,j})| + \frac{1}{\sqrt{2}}(K_f + K_g)h = \bar{J}_g^k$$

Bound in ND ($d = N$):

In the k-th cell of center X_p:

$$J_g^k \leq |f(X_p) - g(X_p)| + \frac{\sqrt{N}}{2}(K_f + K_g)h = \bar{J}_g^k$$

Then,

$$J_g \leq \max_k \bar{J}_g^k$$

Breaking free from grids

Main problem: Learning points are rarely structured as a grid

What about learning in the context of Scientific ML?

~~We control the design of experiment so we could build it as a grid~~

~~**Very constraining:**~~

- ~~The DOE should be defined in advance and we could not add points sequentially~~
- ~~Grids suffer from the curse of dimensionality, the number of f evaluations would grow exponentially with d~~
- ~~Monte Carlo is convenient~~

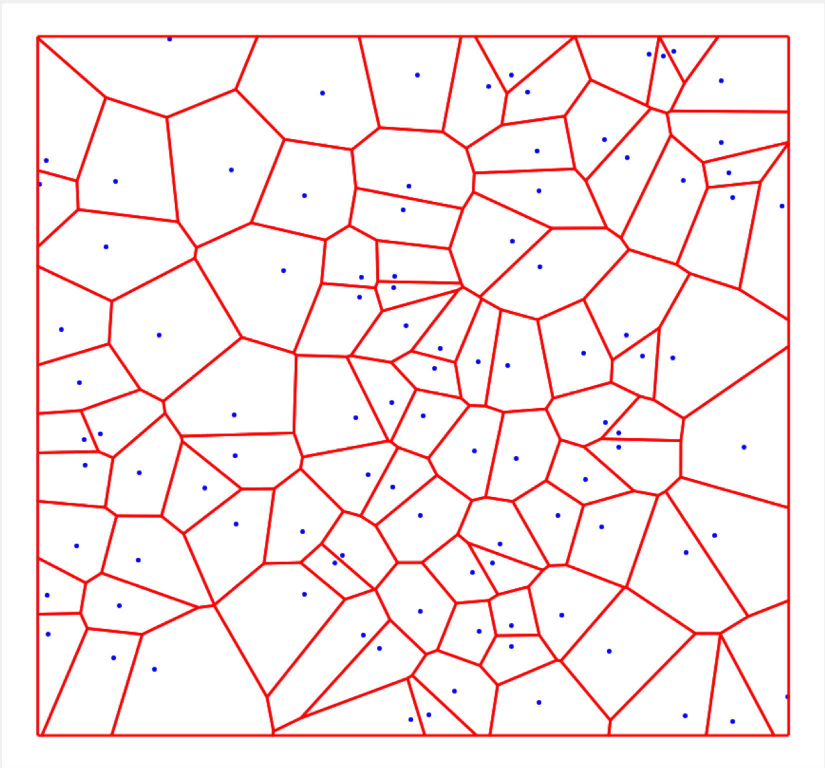
Aim of this work: find ways to build upper bounds for J_g when $\left(X_1, Y_1 = f(X_1)\right), \dots, \left(X_n, Y_n = f(X_n)\right)$ is not structured as a grid

Outline

DEEL

- Introduction
- **Error bound with Voronoï diagrams**
- Bounding with certified Deterministic Optimistic Optimization
- Conclusion & Takeaway

Definition of a Voronoï diagram (and some notations)



A Voronoï diagram $\mathcal{V}^d$ is built on a set of points $\mathbf{X} = \{X_1, \dots, X_n\}, X_i \in \mathcal{X} \subset \mathbb{R}^d$.

Each point is called a site, and the diagram is defined by its cells $\{\mathcal{V}^d(X_1), \dots, \mathcal{V}^d(X_n)\}$ themselves defined by

$$\mathcal{V}^d(X_i) = \{x \in \mathcal{X} \mid \forall j \in \{1, \dots, n\}, \|x - X_i\| \leq \|x - X_j\|\}$$

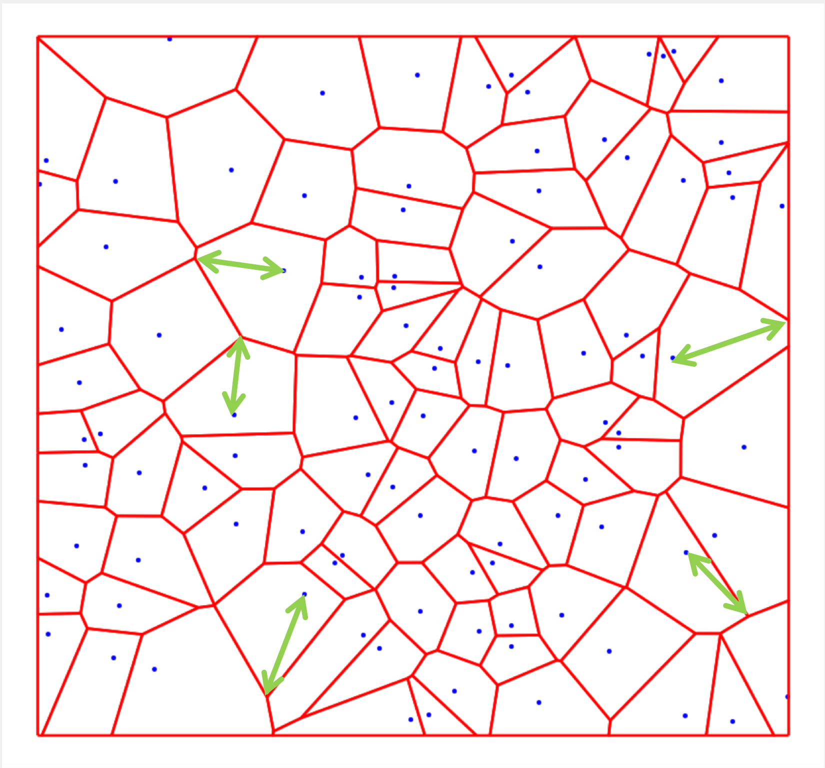
If $x \in \mathcal{V}^d(X_i)$, then X_i is the nearest neighbor of x

We have that $\mathcal{X} = \bigcup_{i \in \{1,\dots,n\}} \mathcal{V}^d(X_i)$, so to obtain $\bar{J}_g$, it is enough finding $\bar{J}_g^i$, an upper bound for

$$J_g^i = \max_{x \in \mathcal{V}^d(X_i)} |f(x) - g(x)|$$

Error bound using Voronoï diagram

DEEL



Let $\color{red}{N}: x \rightarrow \underset{X_i \in \mathbf{X}}{argmin} \|x - X_i\|$ (nearest neighbor map)

Then by the Lipschitz property of g and f, we have that $\forall x \in \mathcal{X}$,

$$|f(x) - g(x)| \leq (K_f + K_g)\|x - N(x)\| + |f(N(x)) - g(N(x))|$$

Lemma 1

Goes well with Voronoï diag!

Let $r(X_i)$ be the radius of $\mathcal{V}^d(X_i)$ defined by

$$r(X_i) = \max_{x \in \mathcal{V}^d(X_i)} \|x - X_i\|$$

Then, it holds that

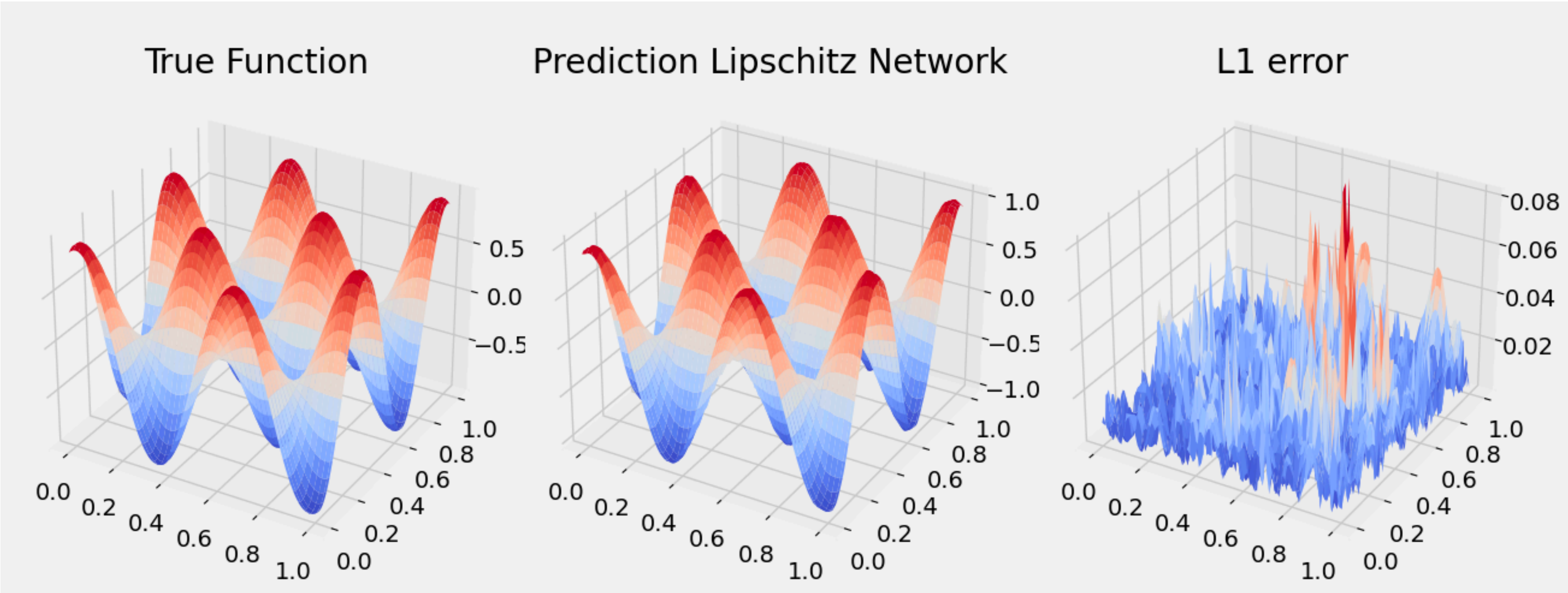
$$J_g^i \leq |f(X_i) - g(X_i)| + (K_f + K_g) r(X_i)$$

Hence,

$$J_g \leq \max_{i \in \{1,\ldots,n\}} |f(X_i) - g(X_i)| + (K_f + K_g) r(X_i)$$

- All we need is to compute $r(X_i)$

Experiments on toy functions

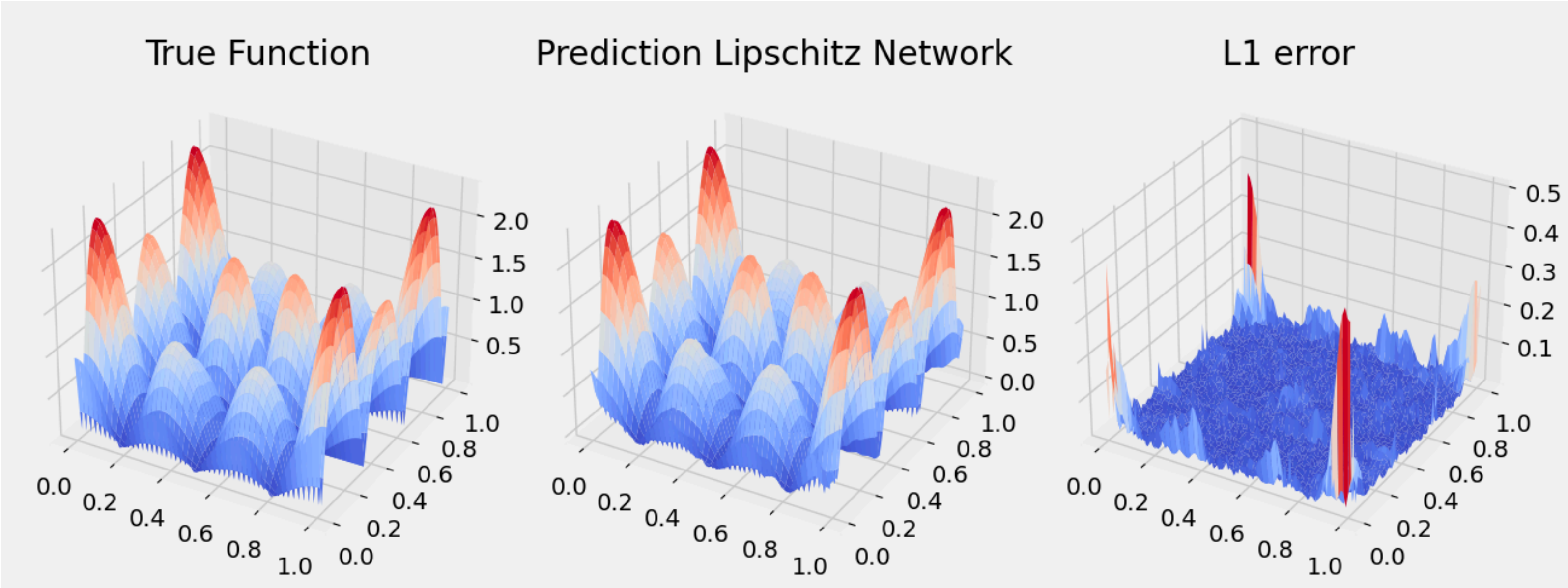


Sinus function

$$f: x, y \to \sin(x) \times \sin(y)$$

10000 training points

Experiments on toy functions



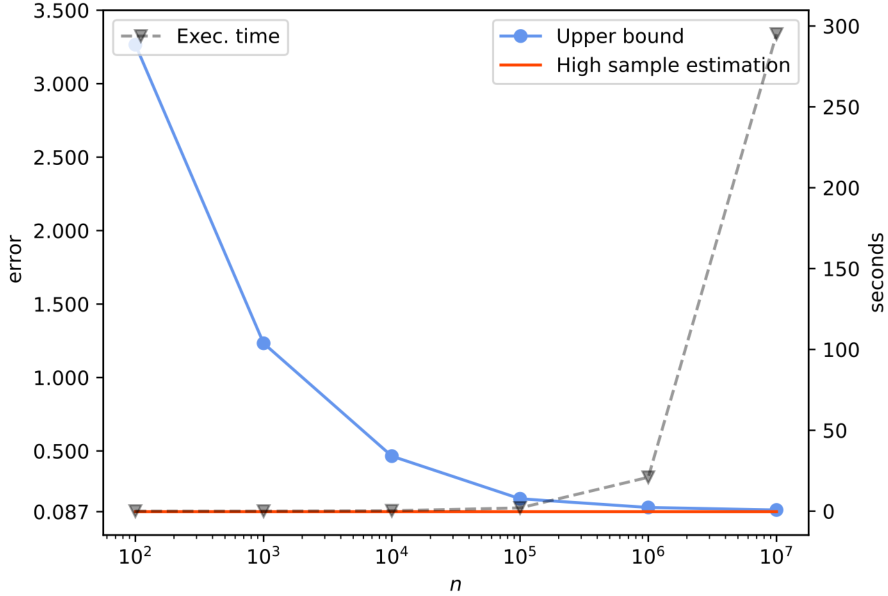
Holder table function

$$f: x, y \to \left| \sin(x) \cos(y) \exp\left(\left| 1 - \frac{\sqrt{x^2 + y^2}}{\pi} \right| \right) \right|$$

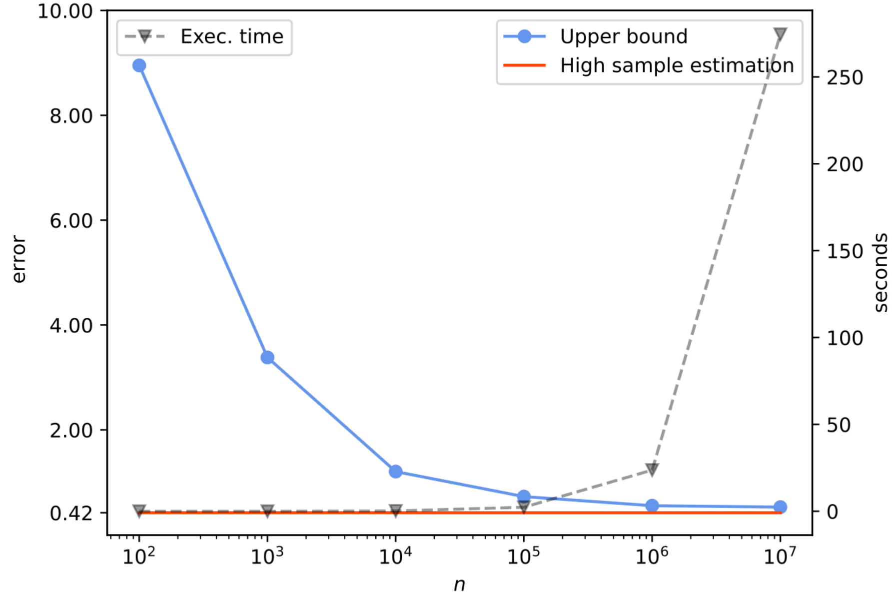
10000 training points

Complexity of Voronoï diagrams

Upper bound of L_∞ error with computation time for Sinus function (left) and Holder table function (right)



Best upper bound: 0.098
High sample estimation: 0.087



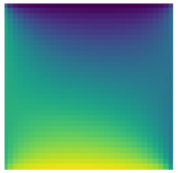
Best upper bound: 0.53
High sample estimation: 0.42

Problem: Voronoï diagram's complexity is exponential**…**

… what about higher d and n?

Diffusion in 2D:

$$\frac{\partial u}{\partial t} = D\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$



- We simulate heat diffusion on a homogeneous surface , with 4 Dirichlet boundary conditions and observe the field at convergence.
- The final heat field depends on the boundary conditions, but not on the initial state nor the diffusivity.

Design of experiment:

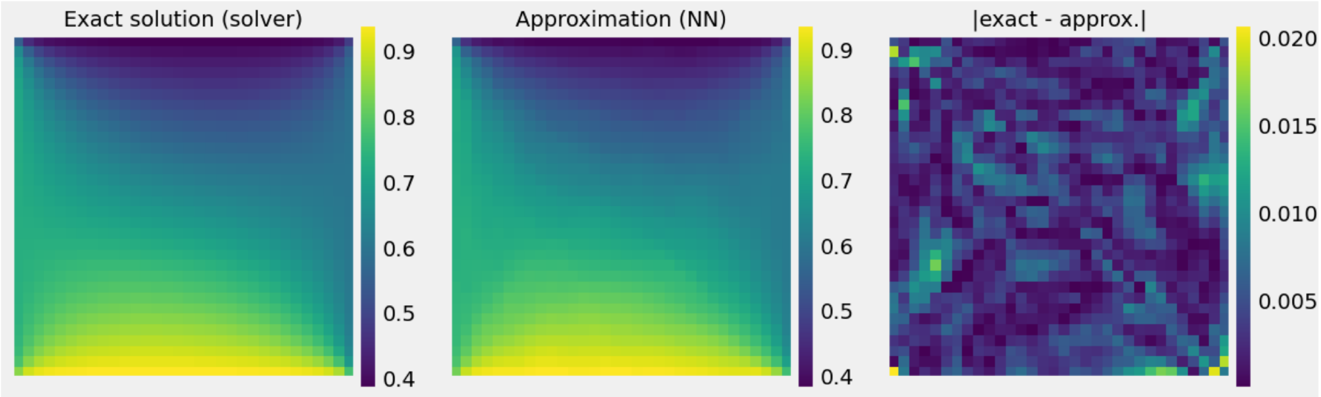
- Sample $n = 5000$ boundary conditions $\{(a_i, b_i, c_i, d_i)\}_{i\in\{1,\ldots,n\}}$ uniformly on $[0,1]^4$.
- Conduct n simulations on a $p\times p$ grid ($p = 32$), yielding a temperature field $\{T_{jk}\}_{j,k\in\{1,\ldots,p\}^2}$.

Training dataset:

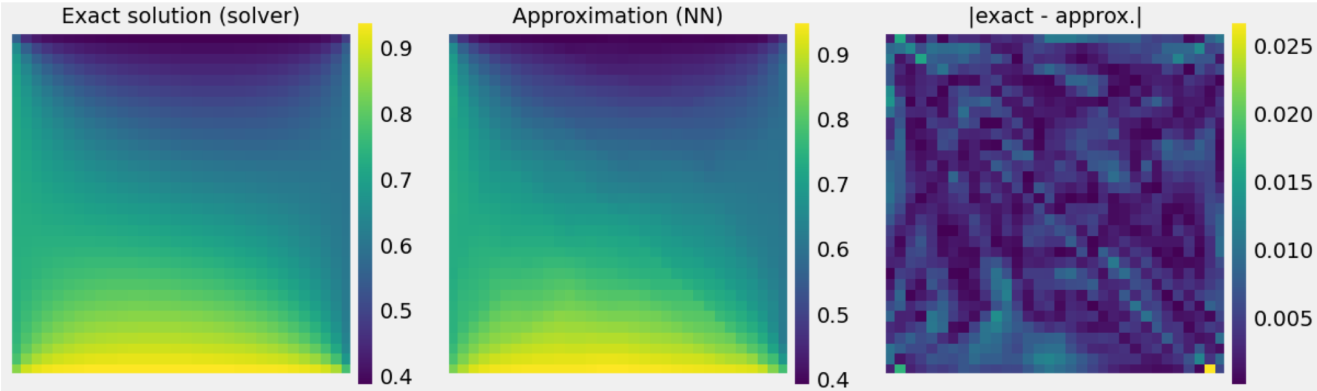
- A subset of $n\times p\times p/10 = 512{,}000$ points $\{(a_i, b_i, c_i, d_i, x_j, x_k), T_{j,k}\}_{i\in\{1,\ldots,n\},j,k\in\{1,\ldots,p\}^2}$

Neural implicit representation approach!

Approximation results



Lipschitz network, MSE=6.3×10^{-5}



Standard fully connected, MSE=4.1×10^{-5}

How to handle unknown K_f?

Two ways:

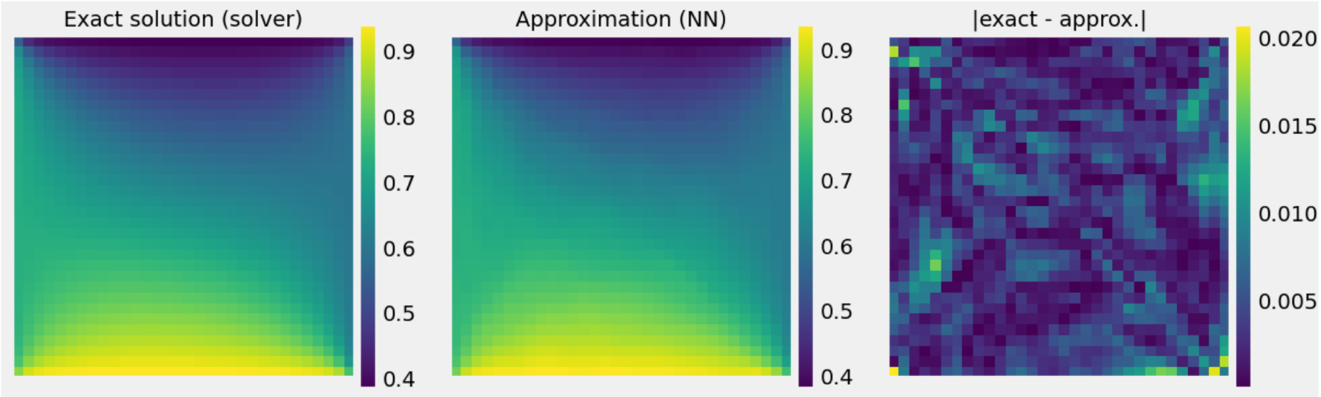
1. Empirical estimation of Lipschitz constant using:

$$\widehat{K_f} = \max_{i \in \{1,\dots,n\}} \left(\max_{X \in \mathcal{N}_k(X_i)} \frac{|f(X) - f(X_i)|}{\|X - X_i\|} \right)$$

   Where $\mathcal{N}_k(X_i)$ is the set of the k-th nearest neighbors of X_i.

2. Hypotheses of f:
   - In [8], the authors compute the Lipschitz constant of f when it is a Gaussian Process interpolating the data.
   - Could apply to polynomial regression
   - We might find the Lipschitz constant by studying the physics [4]

Error bound



Lipschitz network, MSE=6.3×10^{-5}

- Maximum empirical L_1error: $\mathbf{0.17}$

Voronoï diagram with a subset of 20000 points. Takes $\approx$ 3000 seconds ($exponential$ complexity…)

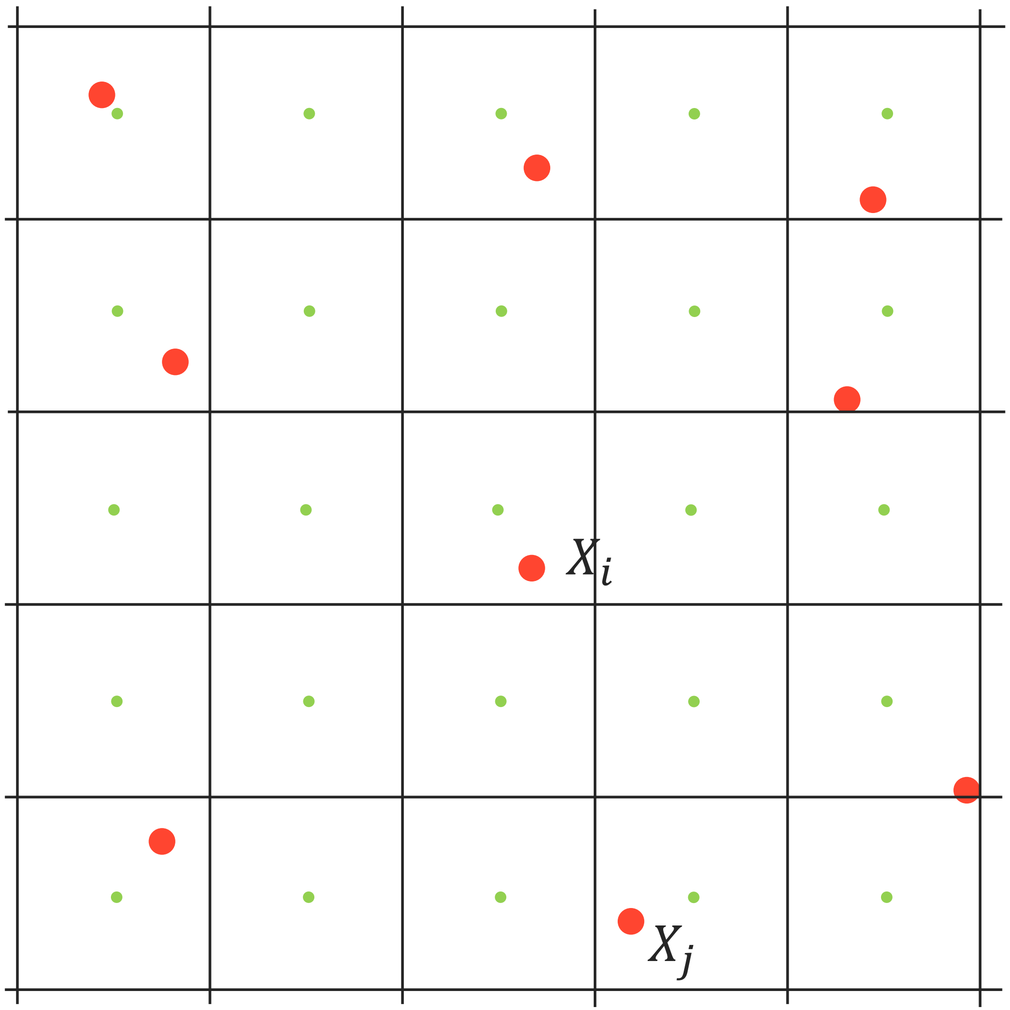
- Error bound: **84**!! Not very appealing…
- We have to find workarounds to use all the $n\times p\times p$ = 5,120,000 points

Outline

- Introduction
- Error bound with Voronoï diagrams
- **Bounding with certified Deterministic Optimistic Optimization**
- Conclusion & Takeaway

Mapping to grid

DE≡L



Let's consider $\mathbf{X} = \{X_1, \ldots, X_n\}$ uniformly distributed on $[0,1]^d$.

We have that, $\forall x, i \in [0,1]^d \times \{1, \ldots, n\}$,

$$|f(x) - g(x)| \leq (K_f + K_g)\|x - X_i\| + |f(X_i) - g(X_i)|$$

We can do better because **we can evaluate $g(x)$**!

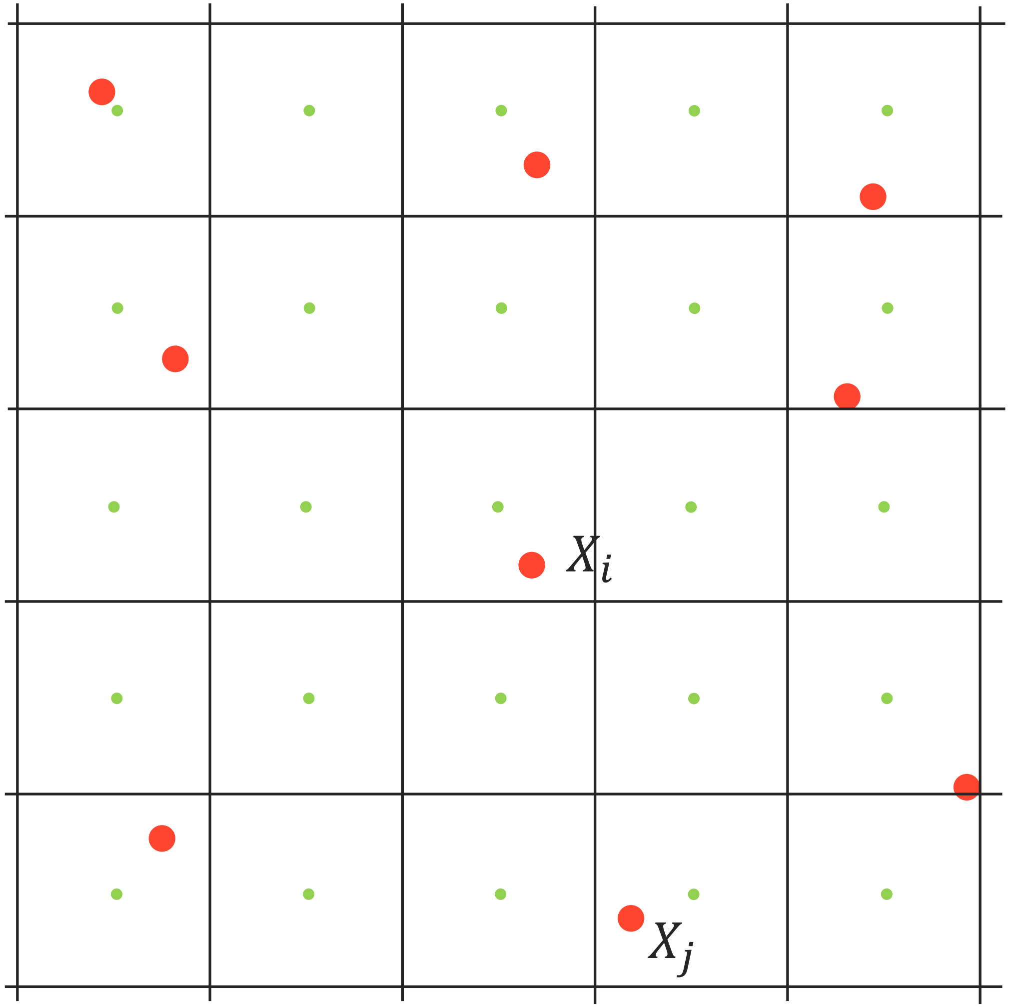
$\forall x, i \in [0,1]^d \times \{1, \ldots, n\}$,

$$|f(x) - g(x)| \leq K_f\|x - X_i\| + |f(X_i) - g(x)|$$

Lemma 2

Now, consider a grid of p^d cells with centers $\{c_1, \ldots, c_{p^d}\}$

Mapping to grid



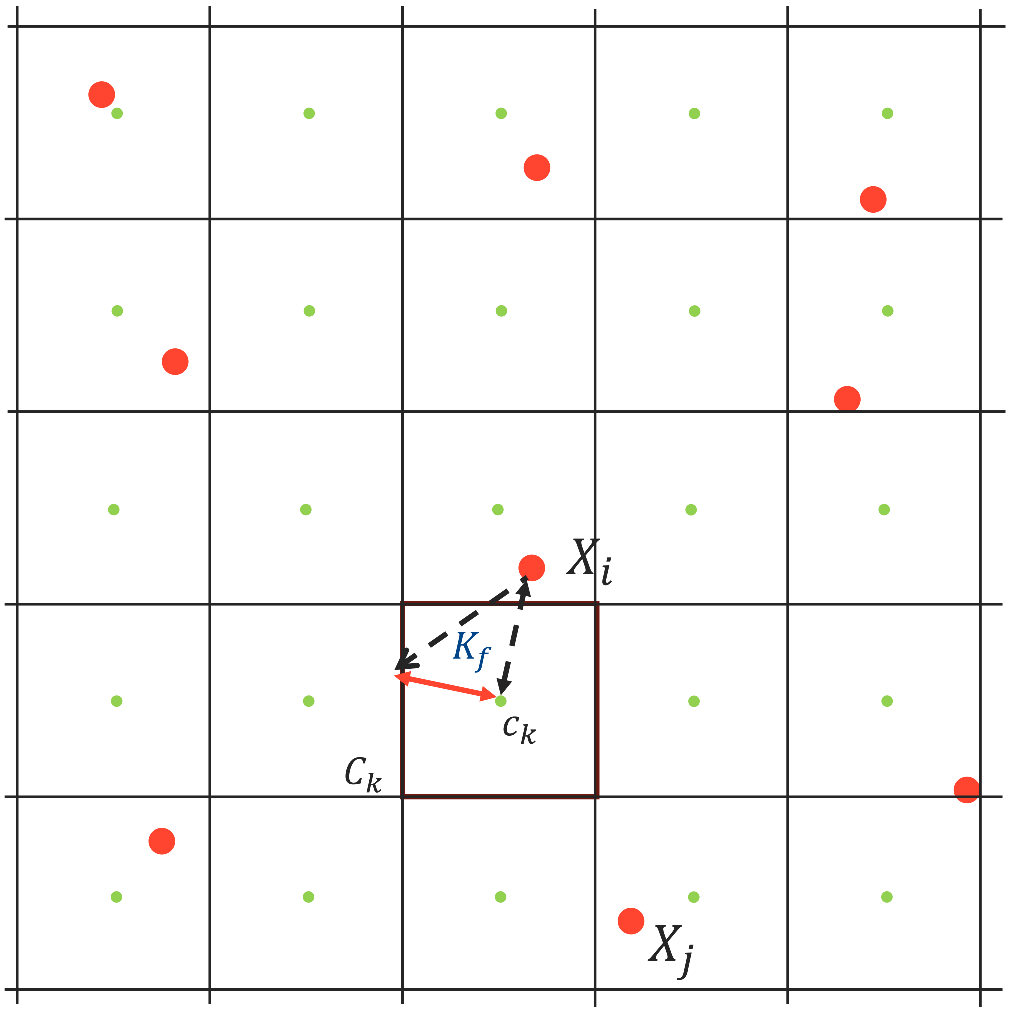
Now, consider a grid of p^d cells $\{C_1, \ldots, C_{p^d}\}$ with centers $\{c_1, \ldots, c_{p^d}\}$

$\forall k \in \{1, \ldots, p^2\}$,

$$|f(c_k) - g(c_k)| \leq K_f \|c_k - X_i\| + |g(c_k) - f(X_i)|$$

Mapping to grid

DEEL

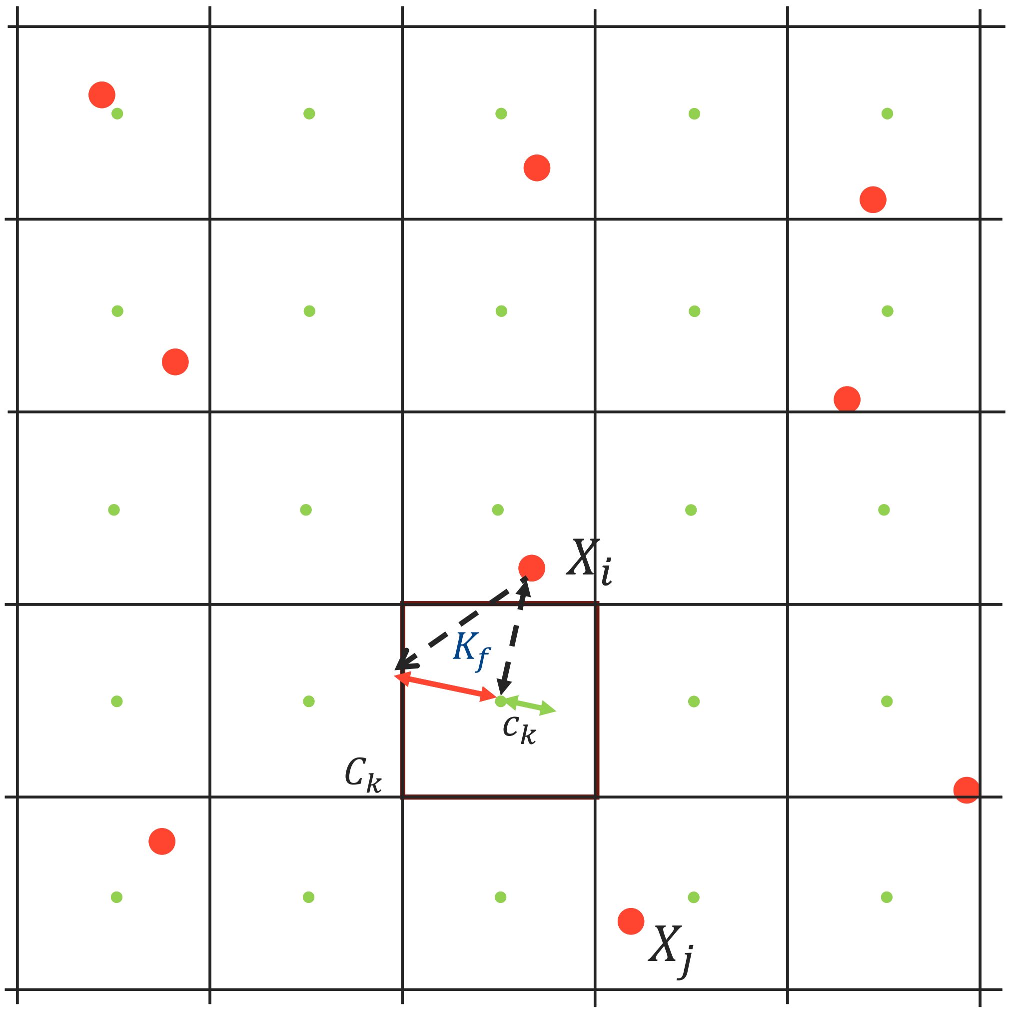


Now, consider a grid of p^d cells $\{C_1, \ldots, C_{p^d}\}$ with centers $\{c_1, \ldots, c_{p^d}\}$

$\forall k \in \{1, \ldots, p^2\}$,

$$|f(c_k) - g(c_k)| \leq K_f \|c_k - X_i\| + |g(c_k) - f(X_i)|$$

Mapping to grid

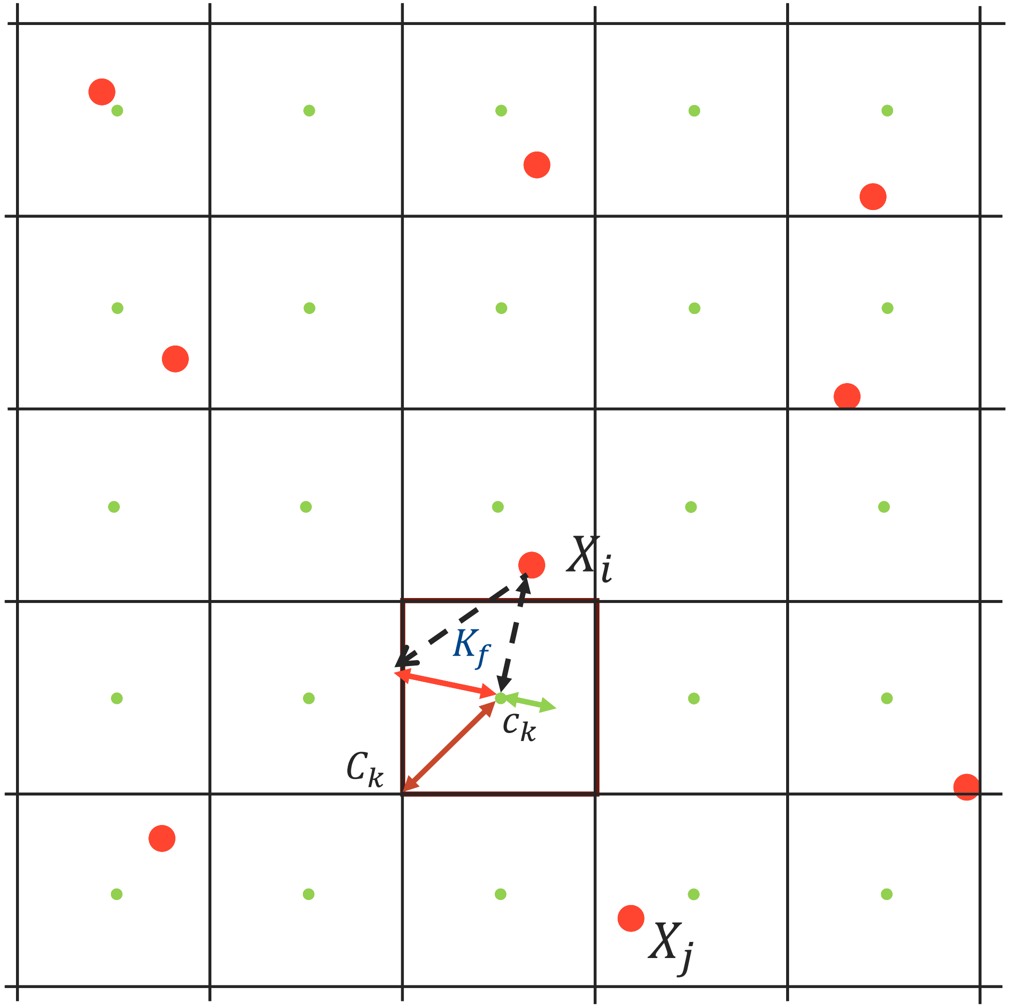


Now, consider a grid of p^d cells $\left\{C_1, \dots, C_{p^d}\right\}$ with centers $\{c_1, \dots, c_{p^d}\}$

$\forall k \in \{1, \dots, p^2\}$,

$$|f(c_k) - g(c_k)| \leq K_f \|c_k - X_i\| + |g(c_k) - f(X_i)|$$

Mapping to grid



Now, consider a grid of p^d cells $\{C_1, \dots, C_{p^d}\}$ with centers $\{c_1, \dots, c_{p^d}\}$

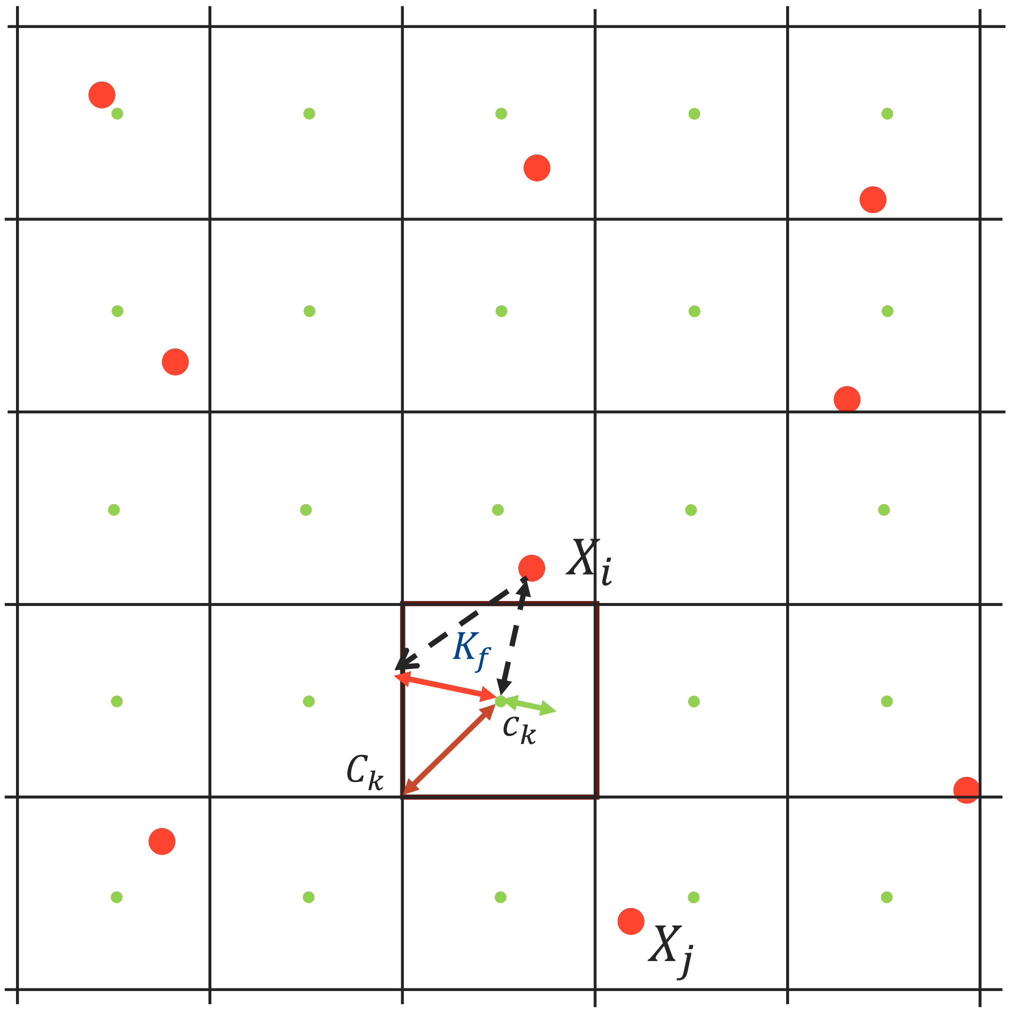
$\forall k \in \{1, \dots, p^2\}$,

$$|f(c_k) - g(c_k)| \le K_f \|c_k - X_i\| + |g(c_k) - f(X_i)|$$

Since we know that $\forall x \in C_k$,

$$|f(x) - g(x)| \le |f(c_k) - g(c_k)| + \frac{\sqrt{d}}{2p}(K_f + K_g)$$

Mapping to grid



Now, consider a grid of p^d cells $\{C_1, \dots, C_{p^d}\}$ with centers $\{c_1, \dots, c_{p^d}\}$

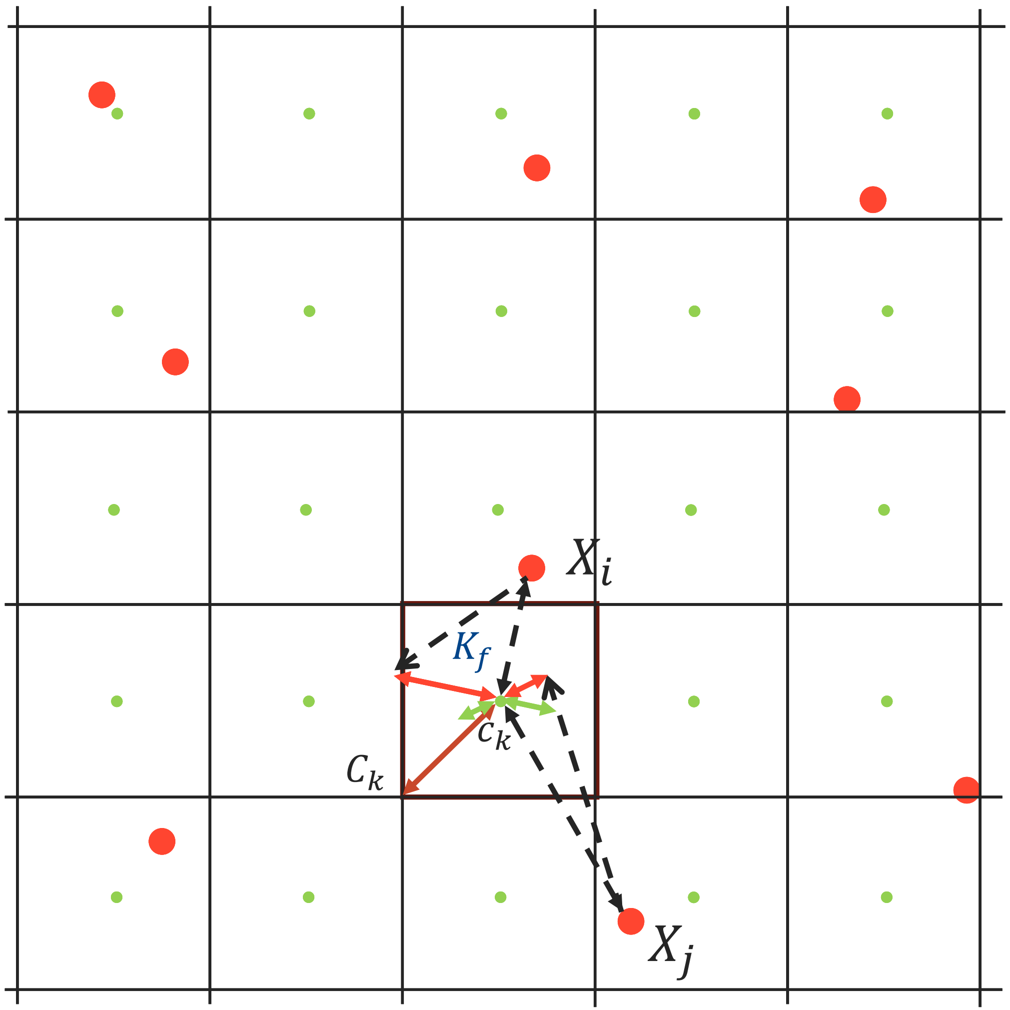
$\forall k \in \{1, \dots, p^2\}$,

$$|f(c_k) - g(c_k)| \leq K_f \|c_k - X_i\| + |g(c_k) - f(X_i)|$$

We have that $\forall x \in C_k$,

$$|f(x) - g(x)| \leq K_f \|c_k - X_i\| + |g(c_k) - f(X_i)| + \frac{\sqrt{d}}{2p}(K_f + K_g)$$

Mapping to grid



Now, consider a grid of p^d cells $\left\{C_1, \dots, C_{p^d}\right\}$with centers $\{c_1, \dots, c_{p^d}\}$

$\forall k \in \{1, \dots, p^2\}$,

$$|f(c_k) - g(c_k)| \leq K_f \|c_k - X_i\| + |g(c_k) - f(X_i)|$$

We have that $\forall x \in C_k$,

$$|f(x) - g(x)| \leq \min_{i \in \{1,\dots,n\}} K_f \|c_k - X_i\| + |g(c_k) - f(X_i)| + \frac{\sqrt{d}}{2p}(K_f + K_g)$$

Mapping to grid

DEEL

$\forall x \in C_k,$

$$|f(x) - g(x)| \leq \underbrace{K_f \|c_k - X_i\|}_{\text{Requires calls to a nearest neighbor algorithm to find } N(c_k)} + \underbrace{|g(c_k) - f(X_i)|}_{\text{Requires evaluations of } g(c_k)\text{, which can be done in batch very efficiently}} + \underbrace{\frac{\sqrt{d}}{2p}(K_f + K_g)}_{\text{free}}$$

Requires calls to a nearest neighbor algorithm to find $N(c_k)$

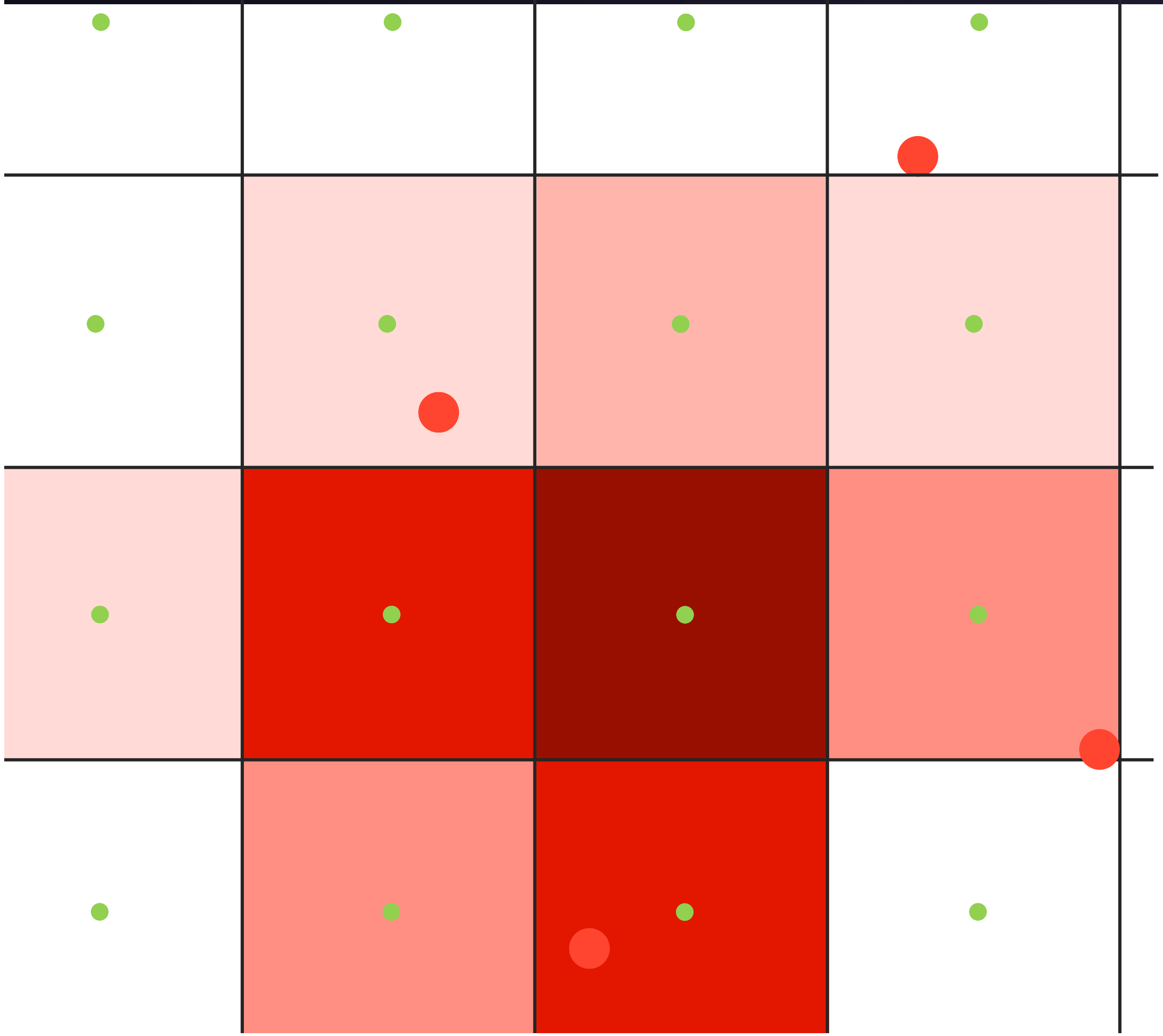
Requires evaluations of $g(c_k)$, which can be done in batch very efficiently

free

Computational efforts needed:

- Nearest neighbor algorithm
  - Many very efficient libraries (immensely cheaper than Voronoï diagram – complexity not exponential)
  - The bound is still valid with approximate nearest neighbors
- Evaluation of g
  - Very efficient on GPU

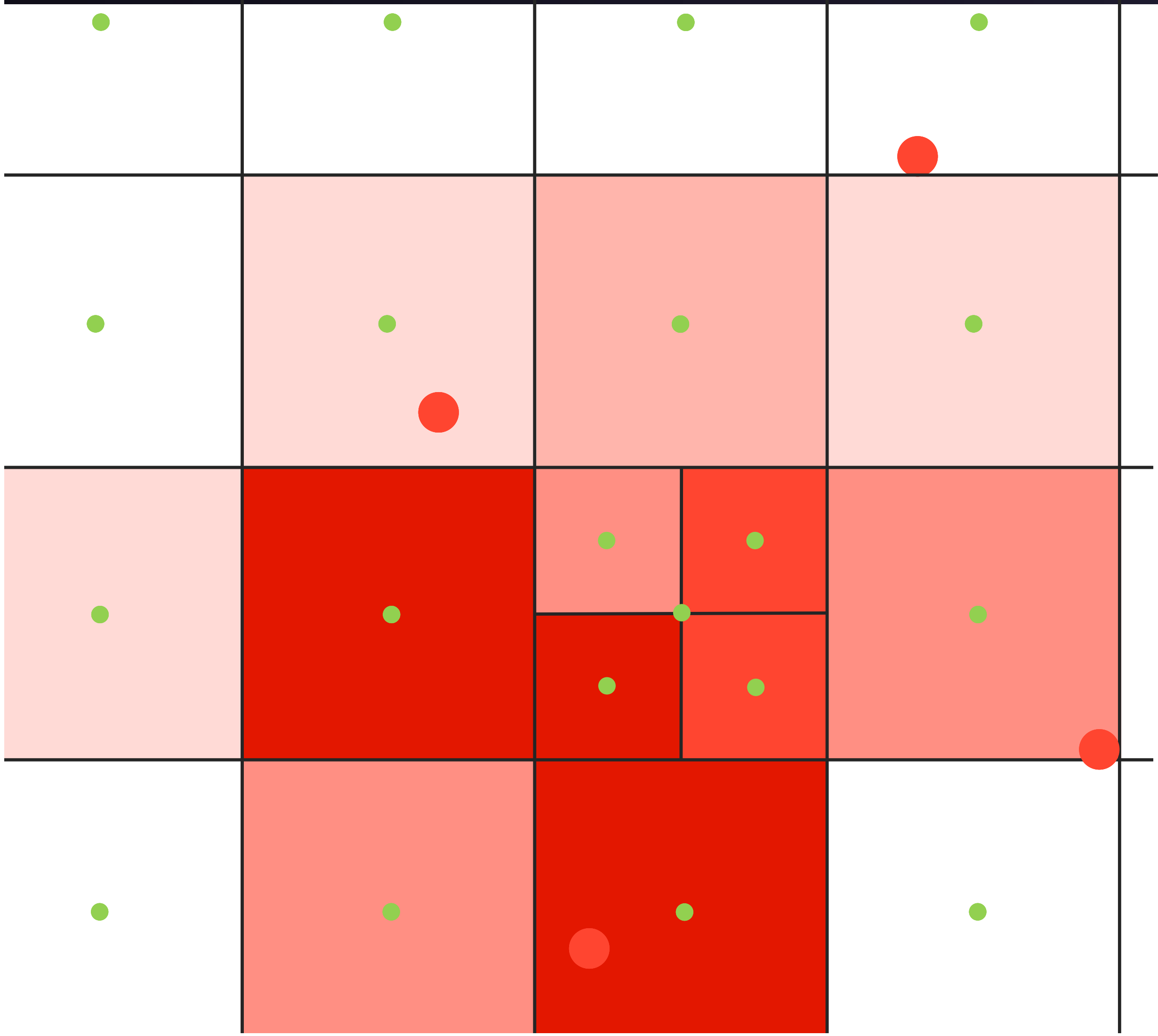
C-DOO



$\forall x \in C_k,$

$$|f(x) - g(x)| \leq \min_{i \in \{1,\dots,n\}} K_f \|c_k - X_i\| + |g(c_k) - f(X_i)| + \frac{\sqrt{d}}{2p}(K_f + K_g)$$

C-DOO



$\forall x \in C_k,$

$$|f(x) - g(x)| \leq \min_{i \in \{1,\ldots,n\}} K_f \|c_k - X_i\| + |g(c_k) - f(X_i)| + \frac{\sqrt{d}}{4p}(K_f + K_g)$$

Certified Deterministic Optimistic Optimization [14]

Split cells until convergence towards

$$\max_{x \in \mathcal{X}} \min_{i \in \{1,\ldots,n\}} K_f \|x - X_i\| + |g(x) - f(X_i)|$$

With a known certificate ϵ

Mapping to grid

DEEL

Certified Deterministic Optimistic Optimization [14]
Finds x^* and ϵ^* such that $\forall x \in \mathcal{X}$

$$\min_{i\in\{1,\ldots,n\}} K_f \|x - X_i\| + |g(x) - f(X_i)| \leq \epsilon^* + \min_{i\in\{1,\ldots,n\}} K_f \|x^* - X_i\| + |g(x^*) - f(X_i)|$$

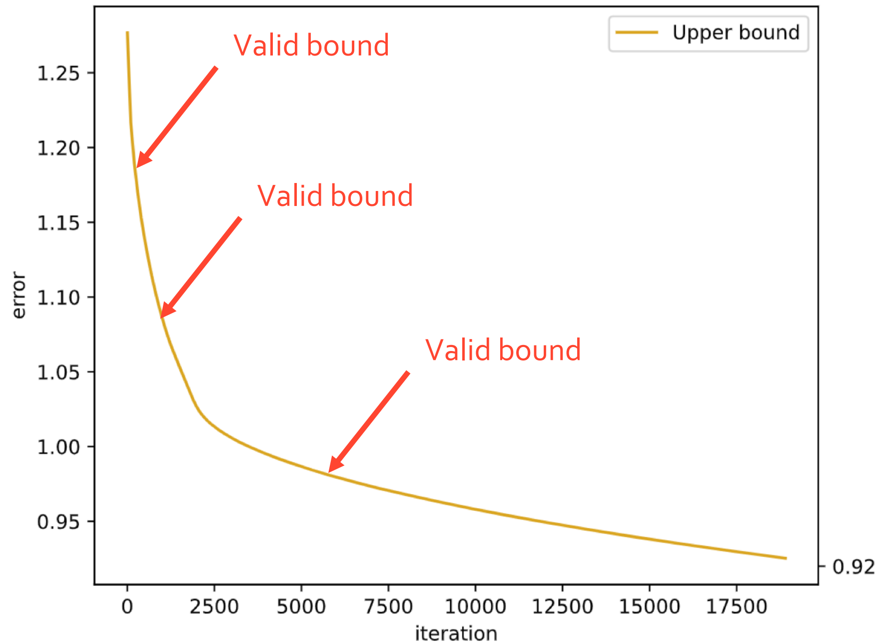
Hence, $\forall x \in \mathcal{X}$

$$|f(x) - g(x)| \leq \epsilon^* + \min_{i\in\{1,\ldots,n\}} K_f \|x^* - X_i\| + |g(x^*) - f(X_i)|$$

New bound !

Results on Heat Diffusion

	Classical Voronoï	Mixed random/mesh	C-DOO
Nb points used	20×10^3	$\mathbf{512\times10^4}$	$\mathbf{512\times10^4}$
Total eval time (sec.)	> 3000	$\mathbf{1.72}$	4,87 per iteration (see graph)
Max L_1 error (est.)	0.17	0.17	0.17
Upper bound	84	1.87	**0, 92**



Can be very long, but can stop anytime to obtain a bound

Beyond SciML: Braking Distance Estimation

DEEL

Braking Distance Estimation for plane landing

	C-DOO
Nb points used	$\mathbf{512 \times 10^4}$
Total eval time (sec.)	7.89
Max L_1 error (est. in meters)	763
Upper bound (in meters)	**1097**

In that case, fast convergence

The bound is not far from the worst error obtained in the training dataset with a Lipschitz neural network.
=> It is practically useful!

Takeaways - conclusions

We built algorithms to compute strict upper bound for $||f - g||_\infty$, where g is a Lipschitz neural net approximating for f. Can be very tight for low dimension.

- Voronoï based, **very costly** because of Voronoï diagram's **exponential complexity**.
- Can be made **way cheaper** by leveraging the **mesh structure** of some data dimensions.
- Can be **relaxed** by casting bounding into an optimization problem and using **C-DOO.**

Perspectives:

- The method is applicable to **any K-lip model** like Gaussian Processes [8] or Polynomial interpolation.
- The algorithms make it possible **to locate the error**, which could be useful for **active learning** (we could provably reduce the error bound) or **sequential optimization**.
- Goes well with the **Neural Implicit Representation** approach (including PINNs).
- **Hybridization** between ML and classical solvers
- The work will continue in ANITI's integrative programs:

DEEL (2) AI4SAVE

Check out "Accelerating hypersonic reentry simulations using deep learning-based hybridization (with guarantees)" Novello et al, Journal of Computational Physics!

Preprint version: available soon. Reach me out to know more and stay up to date.

References

1. Anil, Cem, James Lucas, and Roger Grosse. "Sorting out Lipschitz Function Approximation."ICML, June 11, 2019. https://doi.org/10.48550/arXiv.1811.05381.
2. Baker, Nathan, Frank Alexander, Timo Bremer, Aric Hagberg, Yannis Kevrekidis, Habib Najm, Manish Parashar, et al. "Workshop Report on Basic Research Needs for Scientific Machine Learning: Core Technologies for Artificial Intelligence," February 2019. https://doi.org/10.2172/1478744.
3. Béthune, Louis, Thibaut Boissin, Mathieu Serrurier, Franck Mamalet, Corentin Friedrich, and Alberto González-Sanz. "Pay Attention to Your Loss: Understanding Misconceptions about 1-Lipschitz Neural Networks."NeurIPS, October 17, 2022. https://doi.org/10.48550/arXiv.2104.05097.
4. Bunin, Gene A., and Grégory François. "Lipschitz Constants in Experimental Optimization." arXiv, January 14, 2017. https://doi.org/10.48550/arXiv.1603.07847.
5. Goswami, Somdatta, Aniruddha Bora, Yue Yu, and George Em Karniadakis. "Physics-Informed Deep Neural Operator Networks." arXiv, July 17, 2022. http://arxiv.org/abs/2207.05748.
6. Karniadakis, George, Yannis Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. "Physics-Informed Machine Learning," May 24, 2021, 1–19. https://doi.org/10.1038/s42254-021-00314-5.
7. Kovachki, Nikola, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. "Neural Operator: Learning Maps Between Function Spaces." arXiv, April 7, 2023. http://arxiv.org/abs/2108.08481.
8. Lederer, Armin, Jonas Umlauft, and Sandra Hirche. "Uniform Error Bounds for Gaussian Process Regression with Application to Safe Control."NeurIPS, December 19, 2019. http://arxiv.org/abs/1906.01376.
9. Li, Yichen, Peter Yichen Chen, Tao Du, and Wojciech Matusik. "Learning Preconditioners for Conjugate Gradient PDE Solvers." In *Proceedings of the 40th International Conference on Machine Learning*, 19425–39. PMLR, 2023. https://proceedings.mlr.press/v202/li23e.html.
10. Serrurier, Mathieu, Franck Mamalet, Thomas Fel, Louis Béthune, and Thibaut Boissin. "On the Explainable Properties of 1-Lipschitz Neural Networks: An Optimal Transport Perspective."NeurIPS, June 22, 2023. https://doi.org/10.48550/arXiv.2206.06854.
11. Serrurier, Mathieu, Franck Mamalet, Alberto Gonzalez-Sanz, Thibaut Boissin, Jean-Michel Loubes, and Eustasio del Barrio. "Achieving Robustness in Classification Using Optimal Transport with Hinge Regularization." In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 505–14. Nashville, TN, USA: IEEE, 2021. https://doi.org/10.1109/CVPR46437.2021.00057.
12. Wang, Ruigang, and Ian Manchester. "Direct Parameterization of Lipschitz-Bounded Deep Networks." In *Proceedings of the 40th International Conference on Machine Learning*, 36093–110. PMLR, 2023. https://proceedings.mlr.press/v202/wang23v.html.
13. Novello, Paul, Gaël Poëtte, David Lugato, Simon Peluchon, and Pietro Marco Congedo. "Accelerating Hypersonic Reentry Simulations Using Deep Learning-Based Hybridization (with Guarantees)." Journal of Computational Physics, September 30, 2022. https://doi.org/10.48550/arXiv.2209.13434.