Geometry and AI

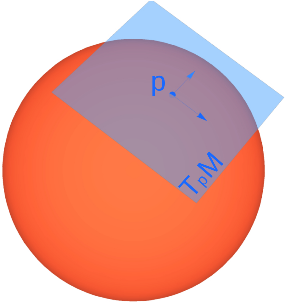
Daniel Platt
AIMS South Africa, 6 May 2025

Geometry is a discipline in pure maths. Geometry and AI have two connections: ”$\rightarrow$”: geometry is used to invent machine learning algorithms, such as geometric kernel methods. ”$\leftarrow$”: machine learning is used to answer questions in geometry, for example by using physics-informed neural networks as a PDE solver.
The structure of the tutorial is: (1) an introduction to Riemannian geometry, (2) positive definite kernels on metric spaces as example of the type ”$\rightarrow$”, (3) a physics-informed neural network on the circle as an example of the type ”$\leftarrow$”.

Tutorial materials, including these slides:
`https://github.com/danielplatt/geometry-ai-tutorial`

Riemannian geometry

- **Manifold**: space like $\{(x, y, z) \in \mathbb{R}^3 : x^2 + y^2 + z^2 - 1 = 0\}$



- p point in M, tangent space:

$$T_pM = \{\text{directions you can go in from } p\}$$

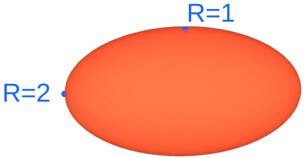
- Riemannian metric, "length of tangent vectors":

$$g : \{\text{tangent vectors}\} \to \mathbb{R}_+$$

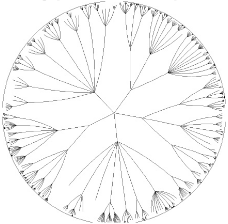
- $\rightsquigarrow$ curvature $R : M \to \mathbb{R}$

Examples:

- Ellipsoid:



- Hyperbolic space:



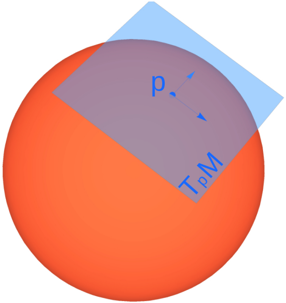
- EEG data:

$$Q = \left\{ A \in \mathbb{R}^{m \times m} : \begin{array}{c} A \text{ symm.} \\ \text{and pos. def.} \end{array} \right\}$$

In neuroscience: ML on Q with different metrics

Riemannian geometry

- Manifold: space like $\{(x,y,z) \in \mathbb{R}^3 : x^2+y^2+z^2-1=0\}$



- p point in M, tangent space:

$$T_pM = \{\text{directions you can go in from } p\}$$

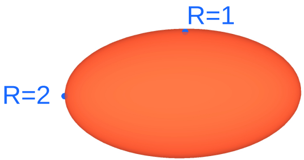
- Riemannian metric, "length of tangent vectors":

$$g : \{\text{tangent vectors}\} \to \mathbb{R}_+$$

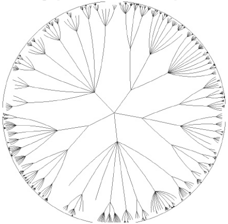
- $\rightsquigarrow$ curvature $R : M \to \mathbb{R}$

Examples:

- Ellipsoid:



- Hyperbolic space:



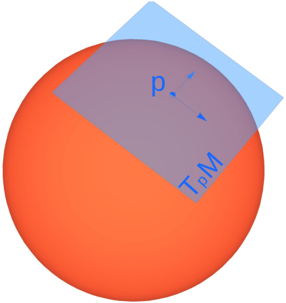
- EEG data:

$$Q = \left\{ A \in \mathbb{R}^{m\times m} : \begin{array}{c} A \text{ symm.} \\ \text{and pos. def.} \end{array} \right\}$$

In neuroscience: ML on Q with different metrics

Riemannian geometry

- Manifold: space like $\{(x, y, z) \in \mathbb{R}^3 : x^2 + y^2 + z^2 - 1 = 0\}$



- p point in M, tangent space:

$$T_pM = \{\text{directions you can go in from } p\}$$

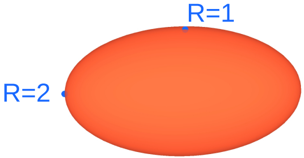
- Riemannian metric, "length of tangent vectors":

$$g : \{\text{tangent vectors}\} \to \mathbb{R}_+$$

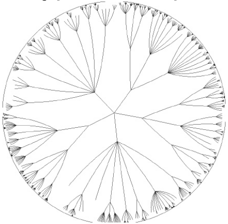
- $\rightsquigarrow$ curvature $R : M \to \mathbb{R}$

Examples:

- Ellipsoid:



- Hyperbolic space:



- EEG data:

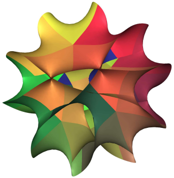
$$Q = \left\{ A \in \mathbb{R}^{m\times m} : \begin{array}{c} A \text{ symm.} \\ \text{and pos. def.} \end{array} \right\}$$

In neuroscience: ML on Q with different metrics

Riemannian geometry

- (Special case of) Yau's theorem: $f(z_1, \ldots, z_4)$ is degree 5 polynomial $\Rightarrow$ there exists a Riemannian metric "Calabi-Yau metric" with curvature $R \equiv 0$ on

$$\{(z_1, \ldots, z_4) \in \mathbb{C}^4 : f(z_1, \ldots, z_4) = 0\}.$$

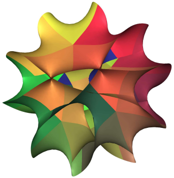


- Question: what does the metric look like?
  Concretely: what's the length of a certain curve? Does the solution to the Poisson problem (defined later) have a zero?
- Best approach to approximately solve the equation $R \equiv 0$ are physics-informed neural networks; approximate solutions are used in two different ways
  1. Inspect approximate solution, formulate conjecture, prove theorem using traditional analysis
  2. Inspect solution, use computer for numerically verified proof
- Several problems in ML can be phrased in Riemannian geometry language

Riemannian geometry

- (Special case of) Yau's theorem: $f(z_1, \ldots, z_4)$ is degree 5 polynomial $\Rightarrow$ there exists a Riemannian metric "Calabi-Yau metric" with curvature $R \equiv 0$ on

$$\{(z_1, \ldots, z_4) \in \mathbb{C}^4 : f(z_1, \ldots, z_4) = 0\}.$$

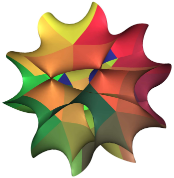


- Question: what does the metric look like?
  Concretely: what's the length of a certain curve? Does the solution to the Poisson problem (defined later) have a zero?
- Best approach to approximately solve the equation $R \equiv 0$ are physics-informed neural networks; approximate solutions are used in two different ways
  1. Inspect approximate solution, formulate conjecture, prove theorem using traditional analysis
  2. Inspect solution, use computer for numerically verified proof
- Several problems in ML can be phrased in Riemannian geometry language

Riemannian geometry

- (Special case of) Yau's theorem: $f(z_1, \ldots, z_4)$ is degree 5 polynomial $\Rightarrow$ there exists a Riemannian metric "Calabi-Yau metric" with curvature $R \equiv 0$ on

$$\{(z_1, \ldots, z_4) \in \mathbb{C}^4 : f(z_1, \ldots, z_4) = 0\}.$$

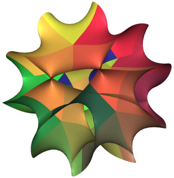


- Question: what does the metric look like?
  Concretely: what's the length of a certain curve? Does the solution to the Poisson problem (defined later) have a zero?
- Best approach to approximately solve the equation $R \equiv 0$ are physics-informed neural networks; approximate solutions are used in two different ways
  1. Inspect approximate solution, formulate conjecture, prove theorem using traditional analysis
  2. Inspect solution, use computer for numerically verified proof
- Several problems in ML can be phrased in Riemannian geometry language

Riemannian geometry

- (Special case of) Yau's theorem: $f(z_1, \ldots, z_4)$ is degree 5 polynomial $\Rightarrow$ there exists a Riemannian metric "Calabi-Yau metric" with curvature $R \equiv 0$ on

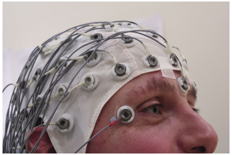
$$\{(z_1, \ldots, z_4) \in \mathbb{C}^4 : f(z_1, \ldots, z_4) = 0\}.$$

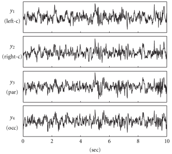


- Question: what does the metric look like?
  Concretely: what's the length of a certain curve? Does the solution to the Poisson problem (defined later) have a zero?
- Best approach to approximately solve the equation $R \equiv 0$ are physics-informed neural networks; approximate solutions are used in two different ways
  1. Inspect approximate solution, formulate conjecture, prove theorem using traditional analysis
  2. Inspect solution, use computer for numerically verified proof
- Several problems in ML can be phrased in Riemannian geometry language

— Blackboard presentation for Riemannian metrics here —

Example 1: positive definite kernels





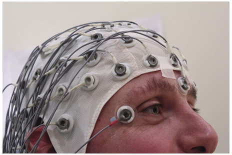
$A \in \mathcal{P} := \{M \in \mathbb{R}^{n\times n} : M$ symmetric and pos.

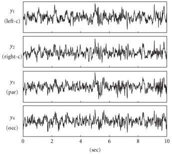
- Goal: classify patient $x_1, \dots, x_k \in \mathcal{P}$ EEG readings (e.g. diseased/non-diseased)
- Riemannian metric g on $\mathcal{P}$ and $\lambda > 0 \Rightarrow$ Gaussian kernel $k : \mathcal{P} \times \mathcal{P} \to \mathbb{R} \Rightarrow$ $\phi : \mathcal{P} \to \mathcal{H}$ and use linear classifier for $\phi(x_1), \dots, \phi(x_k)$ on Hilbert space $\mathcal{H}$
- Different g give different ϕ (Log-Euclidean, Affine-invariant, ...)
- Gaussian kernel: $k(x, y) = \exp(-d(x, y)^2/(2\lambda))$
- Positive definite :$\Leftrightarrow$

$$(k_{ij}) = (k(x_i, x_j)) \text{ is positive semidefinite for all } x_i \in \mathcal{P}.$$

- Heuristic: classification works well if kernel is pos. def.
- Theorem: If k is pos. def. for all $\lambda > 0$, then the space has curvature $R = 0$.
  Theorem: If ex. isometric $S^1 \hookrightarrow \mathcal{P}$, then k not pos. def. for any $\lambda > 0$.
- Open question: ex. space on which k pos. def. for some λ?

Example 1: positive definite kernels





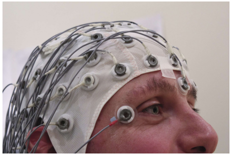
$A \in \mathcal{P} := \{M \in \mathbb{R}^{n\times n} : M$ symmetric and pos.

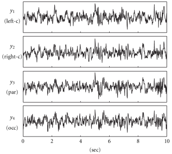
- Goal: classify patient $x_1, \ldots, x_k \in \mathcal{P}$ EEG readings (e.g. diseased/non-diseased)
- Riemannian metric g on $\mathcal{P}$ and $\lambda > 0 \Rightarrow$ Gaussian kernel $k : \mathcal{P} \times \mathcal{P} \to \mathbb{R} \Rightarrow$ $\phi : \mathcal{P} \to \mathcal{H}$ and use linear classifier for $\phi(x_1), \ldots, \phi(x_k)$ on Hilbert space $\mathcal{H}$
- Different g give different ϕ (Log-Euclidean, Affine-invariant, ...)
- Gaussian kernel: $k(x, y) = \exp(-d(x, y)^2/(2\lambda))$
- Positive definite :$\Leftrightarrow$

$$(k_{ij}) = (k(x_i, x_j)) \text{ is positive semidefinite for all } x_i \in \mathcal{P}.$$

- Heuristic: classification works well if kernel is pos. def.
- Theorem: If k is pos. def. for all $\lambda > 0$, then the space has curvature $R = 0$.
  Theorem: If ex. isometric $S^1 \hookrightarrow \mathcal{P}$, then k not pos. def. for any $\lambda > 0$.
- Open question: ex. space on which k pos. def. for some λ?

Example 1: positive definite kernels





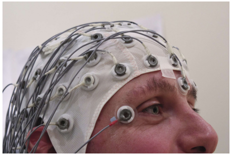
$A \in \mathcal{P} := \{M \in \mathbb{R}^{n \times n} : M$ symmetric and pos.

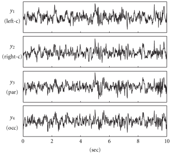
- Goal: classify patient $x_1, \ldots, x_k \in \mathcal{P}$ EEG readings (e.g. diseased/non-diseased)
- Riemannian metric g on $\mathcal{P}$ and $\lambda > 0 \Rightarrow$ Gaussian kernel $k : \mathcal{P} \times \mathcal{P} \to \mathbb{R} \Rightarrow$ $\phi : \mathcal{P} \to \mathcal{H}$ and use linear classifier for $\phi(x_1), \ldots, \phi(x_k)$ on Hilbert space $\mathcal{H}$
- Different g give different ϕ (Log-Euclidean, Affine-invariant, ...)
- Gaussian kernel: $k(x, y) = \exp(-d(x, y)^2/(2\lambda))$
- Positive definite :$\Leftrightarrow$

$$(k_{ij}) = (k(x_i, x_j)) \text{ is positive semidefinite for all } x_i \in \mathcal{P}.$$

- Heuristic: classification works well if kernel is pos. def.
- Theorem: If k is pos. def. for all $\lambda > 0$, then the space has curvature $R = 0$.
  Theorem: If ex. isometric $S^1 \hookrightarrow \mathcal{P}$, then k not pos. def. for any $\lambda > 0$.
- Open question: ex. space on which k pos. def. for some λ?

Example 1: positive definite kernels





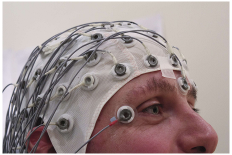
$A \in \mathcal{P} := \{M \in \mathbb{R}^{n\times n} : M$ symmetric and pos.

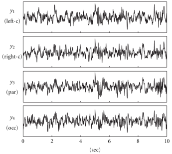
- Goal: classify patient $x_1, \dots, x_k \in \mathcal{P}$ EEG readings (e.g. diseased/non-diseased)
- Riemannian metric g on $\mathcal{P}$ and $\lambda > 0 \Rightarrow$ Gaussian kernel $k : \mathcal{P} \times \mathcal{P} \to \mathbb{R} \Rightarrow$ $\phi : \mathcal{P} \to \mathcal{H}$ and use linear classifier for $\phi(x_1), \dots, \phi(x_k)$ on Hilbert space $\mathcal{H}$
- Different g give different ϕ (Log-Euclidean, Affine-invariant, ...)
- Gaussian kernel: $k(x, y) = \exp(-d(x, y)^2/(2\lambda))$
- Positive definite :$\Leftrightarrow$

$$(k_{ij}) = (k(x_i, x_j)) \text{ is positive semidefinite for all } x_i \in \mathcal{P}.$$

- Heuristic: classification works well if kernel is pos. def.
- Theorem: If k is pos. def. for all $\lambda > 0$, then the space has curvature $R = 0$.
  Theorem: If ex. isometric $S^1 \hookrightarrow \mathcal{P}$, then k not pos. def. for any $\lambda > 0$.
- Open question: ex. space on which k pos. def. for some λ?

Example 1: positive definite kernels





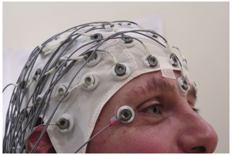
$A \in \mathcal{P} := \{M \in \mathbb{R}^{n\times n} : M$ symmetric and pos.

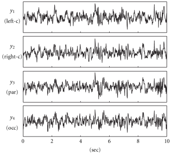
- Goal: classify patient $x_1, \ldots, x_k \in \mathcal{P}$ EEG readings (e.g. diseased/non-diseased)
- Riemannian metric g on $\mathcal{P}$ and $\lambda > 0 \Rightarrow$ Gaussian kernel $k : \mathcal{P} \times \mathcal{P} \to \mathbb{R} \Rightarrow$ $\phi : \mathcal{P} \to \mathcal{H}$ and use linear classifier for $\phi(x_1), \ldots, \phi(x_k)$ on Hilbert space $\mathcal{H}$
- Different g give different ϕ (Log-Euclidean, Affine-invariant, ...)
- Gaussian kernel: $k(x, y) = \exp(-d(x, y)^2/(2\lambda))$
- Positive definite :$\Leftrightarrow$

$$(k_{ij}) = (k(x_i, x_j)) \text{ is positive semidefinite for all } x_i \in \mathcal{P}.$$

- Heuristic: classification works well if kernel is pos. def.
- Theorem: If k is pos. def. for all $\lambda > 0$, then the space has curvature $R = 0$.
  Theorem: If ex. isometric $S^1 \hookrightarrow \mathcal{P}$, then k not pos. def. for any $\lambda > 0$.
- Open question: ex. space on which k pos. def. for some λ?

Example 1: positive definite kernels





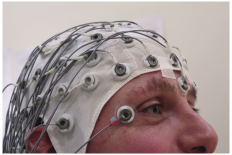
$A \in \mathcal{P} := \{M \in \mathbb{R}^{n\times n} : M$ symmetric and pos. d

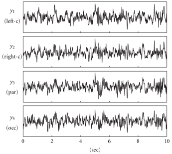
- Goal: classify patient $x_1, \ldots, x_k \in \mathcal{P}$ EEG readings (e.g. diseased/non-diseased)
- Riemannian metric g on $\mathcal{P}$ and $\lambda > 0 \Rightarrow$ Gaussian kernel $k : \mathcal{P} \times \mathcal{P} \to \mathbb{R} \Rightarrow$ $\phi : \mathcal{P} \to \mathcal{H}$ and use linear classifier for $\phi(x_1), \ldots, \phi(x_k)$ on Hilbert space $\mathcal{H}$
- Different g give different ϕ (Log-Euclidean, Affine-invariant, ...)
- Gaussian kernel: $k(x, y) = \exp(-d(x, y)^2/(2\lambda))$
- Positive definite $:\Leftrightarrow$

$$(k_{ij}) = (k(x_i, x_j)) \text{ is positive semidefinite for all } x_i \in \mathcal{P}.$$

- Heuristic: classification works well if kernel is pos. def.
- Theorem: If k is pos. def. for all $\lambda > 0$, then the space has curvature $R = 0$.
  Theorem: If ex. isometric $S^1 \hookrightarrow \mathcal{P}$, then k not pos. def. for any $\lambda > 0$.
- Open question: ex. space on which k pos. def. for some λ?

Example 1: positive definite kernels





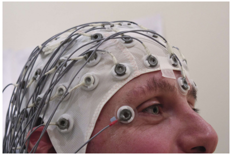
$A \in \mathcal{P} := \{M \in \mathbb{R}^{n\times n} : M$ symmetric and pos.

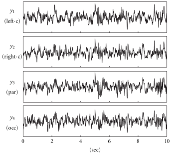
- Goal: classify patient $x_1, \dots, x_k \in \mathcal{P}$ EEG readings (e.g. diseased/non-diseased)
- Riemannian metric g on $\mathcal{P}$ and $\lambda > 0 \Rightarrow$ Gaussian kernel $k : \mathcal{P} \times \mathcal{P} \to \mathbb{R} \Rightarrow$ $\phi : \mathcal{P} \to \mathcal{H}$ and use linear classifier for $\phi(x_1), \dots, \phi(x_k)$ on Hilbert space $\mathcal{H}$
- Different g give different ϕ (Log-Euclidean, Affine-invariant, ...)
- Gaussian kernel: $k(x, y) = \exp(-d(x, y)^2/(2\lambda))$
- Positive definite :$\Leftrightarrow$

$$(k_{ij}) = (k(x_i, x_j)) \text{ is positive semidefinite for all } x_i \in \mathcal{P}.$$

- Heuristic: classification works well if kernel is pos. def.
- Theorem: If k is pos. def. for all $\lambda > 0$, then the space has curvature $R = 0$. Theorem: If ex. isometric $S^1 \hookrightarrow \mathcal{P}$, then k not pos. def. for any $\lambda > 0$.
- Open question: ex. space on which k pos. def. for some λ?

Example 1: positive definite kernels





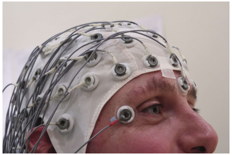
$A \in \mathcal{P} := \{M \in \mathbb{R}^{n\times n} : M$ symmetric and pos.

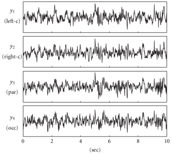
- Goal: classify patient $x_1, \dots, x_k \in \mathcal{P}$ EEG readings (e.g. diseased/non-diseased)
- Riemannian metric g on $\mathcal{P}$ and $\lambda > 0 \Rightarrow$ Gaussian kernel $k : \mathcal{P} \times \mathcal{P} \to \mathbb{R} \Rightarrow$ $\phi : \mathcal{P} \to \mathcal{H}$ and use linear classifier for $\phi(x_1), \dots, \phi(x_k)$ on Hilbert space $\mathcal{H}$
- Different g give different ϕ (Log-Euclidean, Affine-invariant, ...)
- Gaussian kernel: $k(x, y) = \exp(-d(x, y)^2/(2\lambda))$
- Positive definite :$\Leftrightarrow$

$$(k_{ij}) = (k(x_i, x_j)) \text{ is positive semidefinite for all } x_i \in \mathcal{P}.$$

- Heuristic: classification works well if kernel is pos. def.
- Theorem: If k is pos. def. for all $\lambda > 0$, then the space has curvature $R = 0$.
  Theorem: If ex. isometric $S^1 \hookrightarrow \mathcal{P}$, then k not pos. def. for any $\lambda > 0$.
- Open question: ex. space on which k pos. def. for some λ?

Example 1: positive definite kernels





$A \in \mathcal{P} := \{M \in \mathbb{R}^{n\times n} : M$ symmetric and pos.

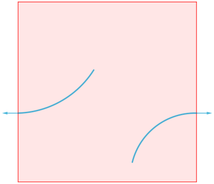
- Goal: classify patient $x_1, \dots, x_k \in \mathcal{P}$ EEG readings (e.g. diseased/non-diseased)
- Riemannian metric g on $\mathcal{P}$ and $\lambda > 0 \Rightarrow$ Gaussian kernel $k : \mathcal{P} \times \mathcal{P} \to \mathbb{R} \Rightarrow$ $\phi : \mathcal{P} \to \mathcal{H}$ and use linear classifier for $\phi(x_1), \dots, \phi(x_k)$ on Hilbert space $\mathcal{H}$
- Different g give different ϕ (Log-Euclidean, Affine-invariant, ...)
- Gaussian kernel: $k(x, y) = \exp(-d(x, y)^2/(2\lambda))$
- Positive definite :$\Leftrightarrow$

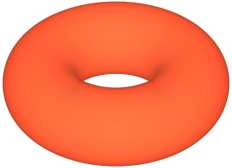
$$(k_{ij}) = (k(x_i, x_j)) \text{ is positive semidefinite for all } x_i \in \mathcal{P}.$$

- Heuristic: classification works well if kernel is pos. def.
- Theorem: If k is pos. def. for all $\lambda > 0$, then the space has curvature $R = 0$.
  Theorem: If ex. isometric $S^1 \hookrightarrow \mathcal{P}$, then k not pos. def. for any $\lambda > 0$.
- Open question: ex. space on which k pos. def. for some λ?

Example 2: the Poisson problem

- 3-torus $([0,1] \times [0,1] \times [0,1])/\sim$, where $\sim$ identifies opposing faces



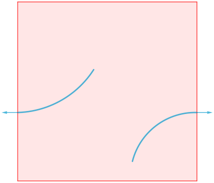


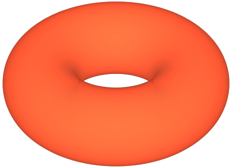
(standard metric) (funny metric)

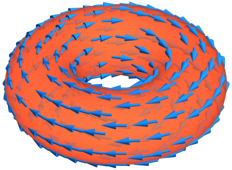
- Vector field: $X : T^3 \to \{\text{tangent vectors}\}$
- Laplace operator $\Delta : \{\text{vector fields}\} \to \{\text{vector fields}\}$
- Question: does there exist a "Calabi-Yau" metric on T^3 such that every vector field X satisfying $\Delta X = 0$ has a zero?
- Easier question: does there exists *any* metric on T^3 such that every (or some) vector field X satisfying $\Delta X = 0$ has a zero?
- Approach: numerically verified proof:
  1. Use physics-informed neural network (PINN) to solve $\Delta X = 0 \to$ approximate solution X_{PINN} with zero
  2. Prove an error bound for residual $|\Delta X_{\mathsf{PINN}}|$ of the PINN solution ($\leftarrow$ exercise)
  3. Conclude that the nearby genuine solution X also must have a zero ($\leftarrow$ pure maths)

Example 2: the Poisson problem

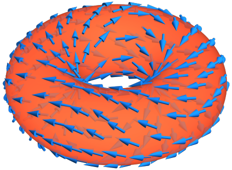
- 3-torus $([0,1] \times [0,1] \times [0,1])/\sim$, where $\sim$ identifies opposing faces







(standard metric)

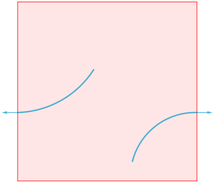
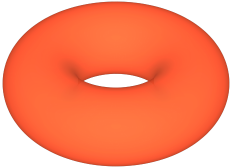
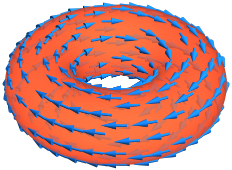


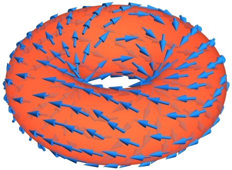
(funny metric)

- Vector field: $X : T^3 \to \{\text{tangent vectors}\}$
- Laplace operator $\Delta : \{\text{vector fields}\} \to \{\text{vector fields}\}$
- Question: does there exist a "Calabi-Yau" metric on T^3 such that every vector field X satisfying $\Delta X = 0$ has a zero?
- Easier question: does there exists *any* metric on T^3 such that every (or some) vector field X satisfying $\Delta X = 0$ has a zero?
- Approach: numerically verified proof:
  1. Use physics-informed neural network (PINN) to solve $\Delta X = 0 \to$ approximate solution X_{PINN} with zero
  2. Prove an error bound for residual $|\Delta X_{\text{PINN}}|$ of the PINN solution ($\leftarrow$ exercise)
  3. Conclude that the nearby genuine solution X also must have a zero ($\leftarrow$ pure maths)

Example 2: the Poisson problem

- 3-torus $([0,1] \times [0,1] \times [0,1])/\sim$, where $\sim$ identifies opposing faces

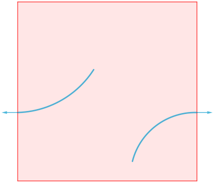



(standard metric)

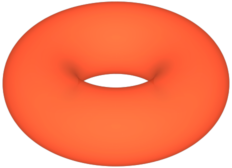

(funny metric)

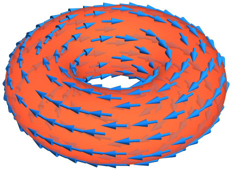
- Vector field: $X : T^3 \to \{\text{tangent vectors}\}$
- Laplace operator $\Delta : \{\text{vector fields}\} \to \{\text{vector fields}\}$
- Question: does there exist a "Calabi-Yau" metric on T^3 such that every vector field X satisfying $\Delta X = 0$ has a zero?
- Easier question: does there exists *any* metric on T^3 such that every (or some) vector field X satisfying $\Delta X = 0$ has a zero?
- Approach: numerically verified proof:
  1. Use physics-informed neural network (PINN) to solve $\Delta X = 0 \to$ approximate solution X_{PINN} with zero
  2. Prove an error bound for residual $|\Delta X_{\text{PINN}}|$ of the PINN solution ($\leftarrow$ exercise)
  3. Conclude that the nearby genuine solution X also must have a zero ($\leftarrow$ pure maths)

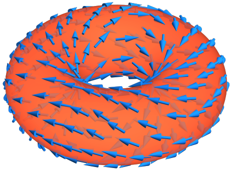
Example 2: the Poisson problem

- 3-torus $([0,1]\times[0,1]\times[0,1])/\sim$, where $\sim$ identifies opposing faces






(standard metric)


(funny metric)

- Vector field: $X : T^3 \to \{\text{tangent vectors}\}$
- Laplace operator $\Delta : \{\text{vector fields}\} \to \{\text{vector fields}\}$
- Question: does there exist a "Calabi-Yau" metric on T^3 such that every vector field X satisfying $\Delta X = 0$ has a zero?
- Easier question: does there exists *any* metric on T^3 such that every (or some) vector field X satisfying $\Delta X = 0$ has a zero?
- Approach: numerically verified proof:
  1. Use physics-informed neural network (PINN) to solve $\Delta X = 0 \to$ approximate solution X_{PINN} with zero
  2. Prove an error bound for residual $|\Delta X_{\text{PINN}}|$ of the PINN solution ($\leftarrow$ exercise)
  3. Conclude that the nearby genuine solution X also must have a zero ($\leftarrow$ pure maths)

Thank you for the attention!

Image credit: Calabi-Yau variety by Andrew J. Hanson - Ticket#2014010910010981, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=30579072
EEG recording cap: A cap holds electrodes in place while recording an EEG. 29 October 2012, 17:44:31. https://www.flickr.com/photos/tim_uk/8135755109/. Chris Hope. CC BY 2.0.
Time series data: Luca Faes, Silvia Erla, Giandomenico Nollo: Measuring Connectivity in Linear Multivariate Processes: Definitions, Interpretation, and Practical Analysis