Learning from scarce data by combining machine learning and fundamental ecological knowledge

University of Fribourg

Victor Boussange

Swiss Federal Research Institute for Forest, Snow and Landscape Research (WSL)

My background

My background

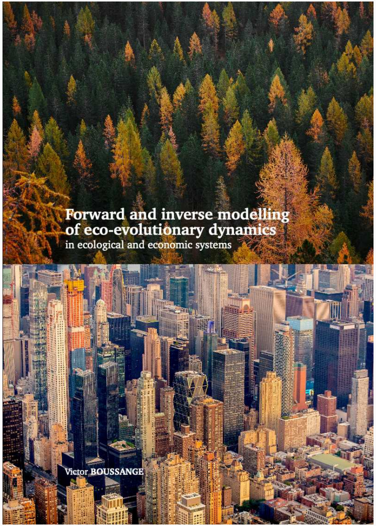
- Born in Bordeaux, France
- Studied in INSA Lyon, France | **Engineering**

My background

- Born in Bordeaux, France
- Studied in INSA Lyon, France | **Engineering**
- Went to Sydney, Australia | **Master thesis in theoretical geomechanics**

My background

- Born in Bordeaux, France
- Studied in INSA Lyon, France | **Engineering**
- Went to Sydney, Australia | **Master thesis in theoretical geomechanics**
- **PhD @ ETH Zürich**, Switzerland in the "Ecosystem and Landscape Evolution" group



My background

- Born in Bordeaux, France
- Studied in INSA Lyon, France | **Engineering**
- Went to Sydney, Australia | **Master thesis in theoretical geomechanics**
- **PhD @ ETH Zürich**, Switzerland in the "Ecosystem and Landscape Evolution" group
- Took a sabbatical, **sailed Tippelei** to Norway

My background

- Born in Bordeaux, France
- Studied in INSA Lyon, France | **Engineering**
- Went to Sydney, Australia | **Master thesis in theoretical geomechanics**
- **PhD @ ETH Zürich**, Switzerland in the "Ecosystem and Landscape Evolution" group
- Took a sabbatical, **sailed Tippelei** to Norway

My background

- Born in Bordeaux, France
- Studied in INSA Lyon, France | **Engineering**
- Went to Sydney, Australia | **Master thesis in theoretical geomechanics**
- **PhD @ ETH Zürich**, Switzerland in the "Ecosystem and Landscape Evolution" group
- Took a sabbatical, **sailed Tippelei** to Norway
- Currently working at **WSL in the Dynamic Macroecology group**
    - Feedbacks

My background

- Born in Bordeaux, France
- Studied in INSA Lyon, France | **Engineering**
- Went to Sydney, Australia | **Master thesis in theoretical geomechanics**
- **PhD @ ETH Zürich**, Switzerland in the ”Ecosystem and Landscape Evolution” group
- Took a sabbatical, **sailed Tippelei** to Norway
- Currently working at **WSL in the Dynamic Macroecology group**
  - Feedbacks
  - Speed2Zero

SPEED2ZERO

My interests

My interests

- What are the processes and mechanisms that drive life on Earth?

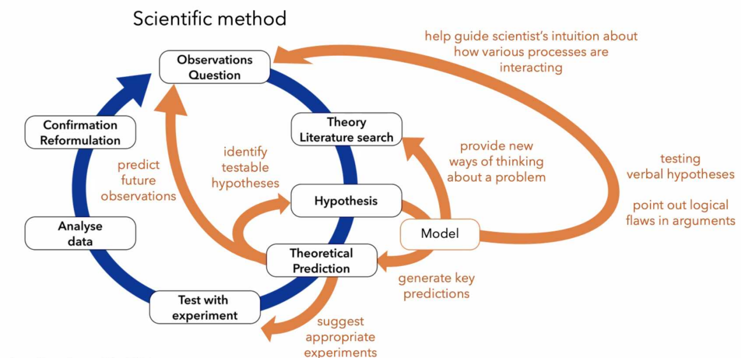
My interests

- What are the processes and mechanisms that drive life on Earth?
- How can we use this knowledge to benefit society?

Modelling in ecology

Modelling in ecology

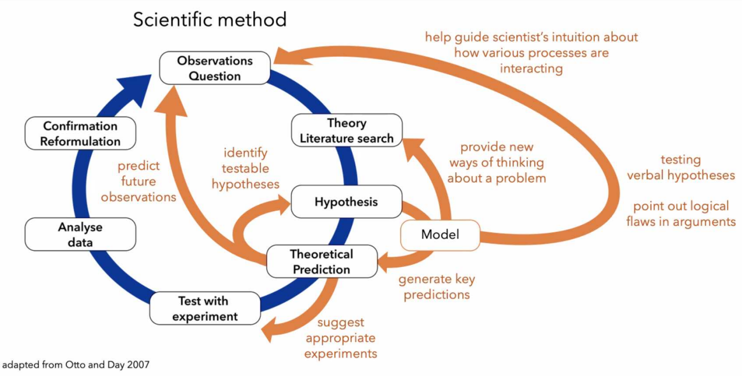
Models to advance ecological theory



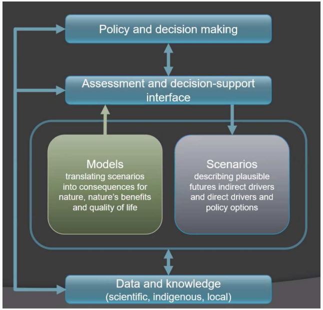
adapted from Otto and Day 2007

Modelling in ecology

Models to advance ecological theory



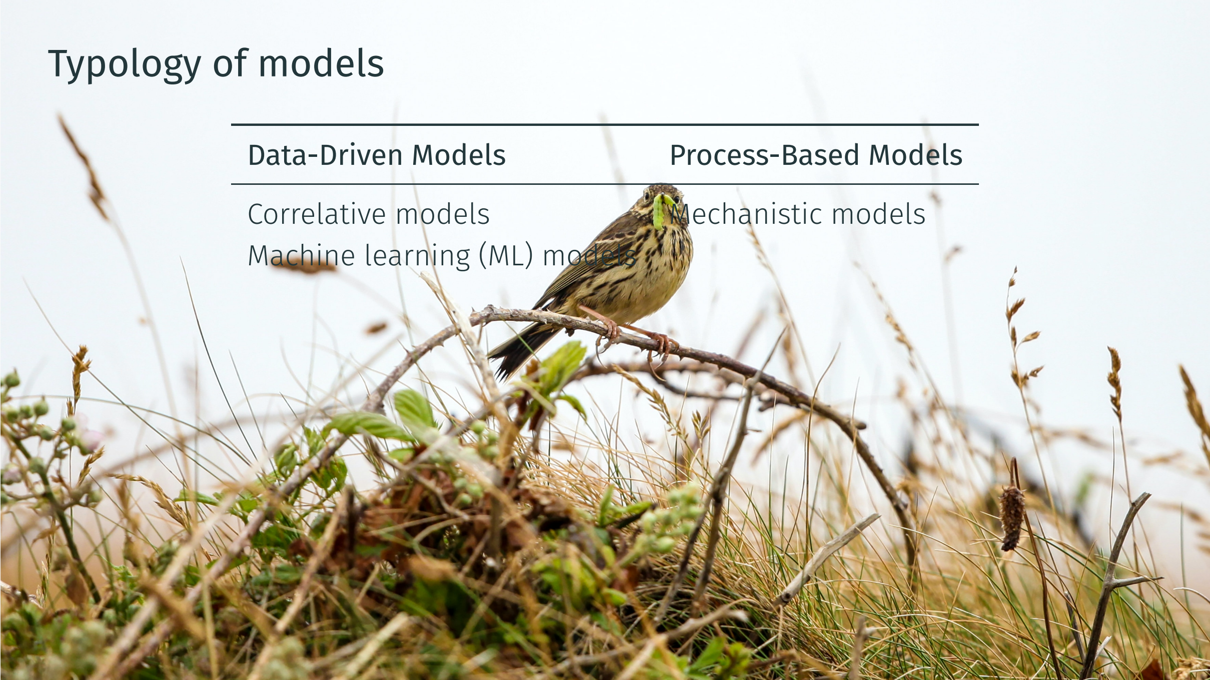
Models are useful for society



Typology of models

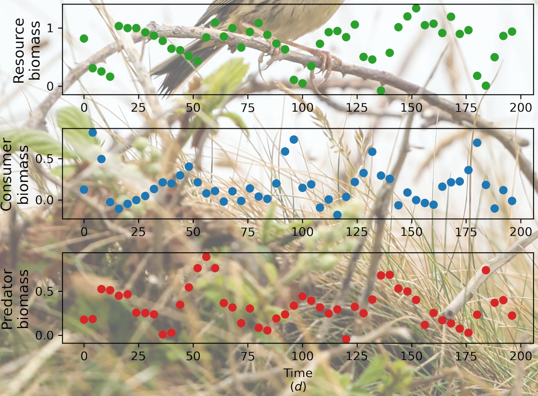
Typology of models

Data-Driven Models	Process-Based Models
Correlative models	Mechanistic models
Machine learning (ML) models	

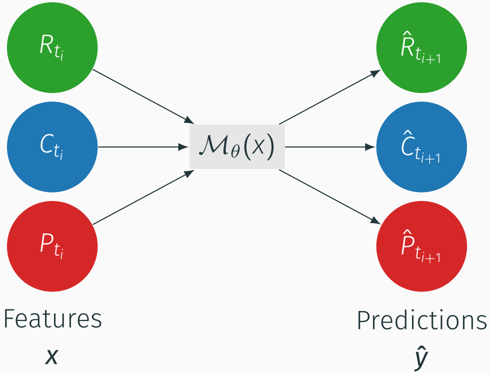


Typology of models

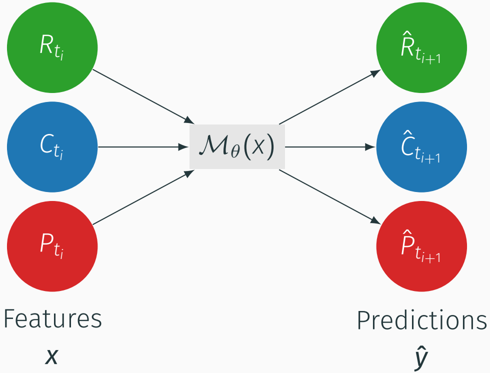
Data-Driven Models	Process-Based Models
Correlative models	Mechanistic models
Machine learning (ML) models	



Data-based modelling



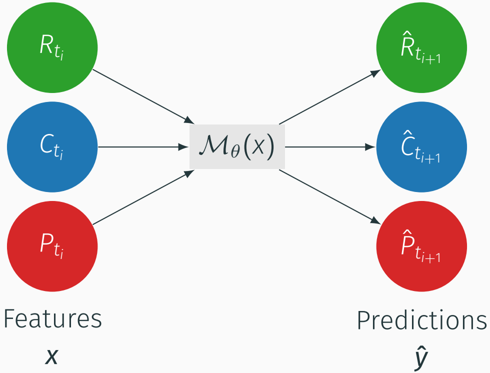
Data-based modelling



Loss function

$$
\begin{aligned}
L(\theta, \mathbf{y}) &= \sum_{k=1}^{K} ||y_k - \mathcal{M}_\theta(x_k)||^2 \\
&\propto -\log p(\mathbf{y}|\theta)
\end{aligned}
$$

Data-based modelling



Loss function

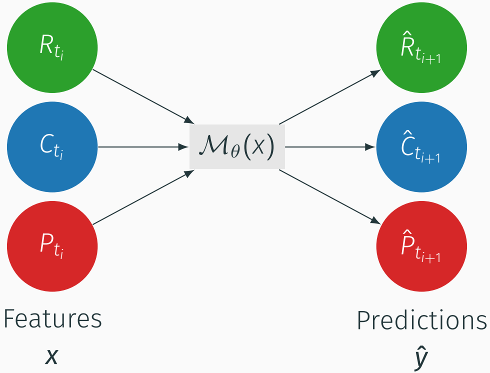
$$L(\theta, \mathbf{y}) = \sum_{k=1}^{K} ||y_k - \mathcal{M}_\theta(x_k)||^2$$

$$\propto -\log p(\mathbf{y}|\theta)$$

Find the parameters $\hat{\theta}$ that minimize the negative logarithm of the posterior

$$\hat{\theta} = \arg\min_\theta L(\theta, \mathbf{y})$$

Data-based modelling



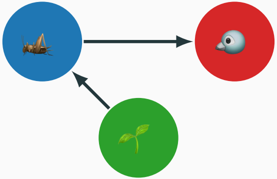
Loss function

$$L(\theta, \mathbf{y}) = \sum_{k=1}^{K} ||y_k - \mathcal{M}_\theta(x_k)||^2$$
$$\propto -\log p(\mathbf{y}|\theta)$$

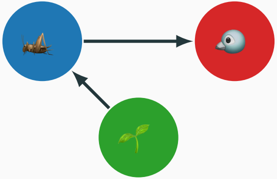
Find the parameters $\hat{\theta}$ that minimize the negative logarithm of the posterior

$$\hat{\theta} = \arg\min_\theta L(\theta, \mathbf{y})$$

Process-based modelling



Process-based modelling

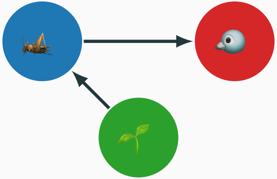


net growth rate 🌱 = basal growth − competition − grazing − death

net rowth rate 🦗 = grazing − predation − death

net growth rate 🐦 = predation − death

Process-based modelling



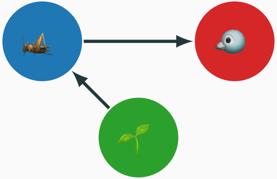
$$\frac{dy}{dt} = f(y, t)$$

$$\frac{d}{dt} R_t = \overbrace{R_t(1 - R_t)}^{\text{logistic growth}} - x_c y_c \underbrace{\frac{C_t R_t}{R_t + R_0}}_{\substack{\text{functional response} \\ \text{(intake rate of consumers)}}}$$

$$\frac{d}{dt} C_t = x_c C_t \left[-1 + y_c \frac{R_t}{R_t + R_0} \right] - x_p y_p \underbrace{\frac{P_t C_t}{C_t + C_0}}_{\substack{\text{functional response} \\ \text{(intake rate of predators)}}}$$

$$\frac{d}{dt} P_t = x_p P_t \left[-1 + y_p \frac{C_t}{C_t + C_0} \right]$$

Process-based modelling



$$\frac{d}{dt} R_t = \overbrace{R_t(1 - R_t)}^{\text{logistic growth}} - x_c y_c \underbrace{\frac{C_t R_t}{R_t + R_0}}_{\substack{\text{functional response} \\ \text{(intake rate of consumers)}}}$$

$$\frac{d}{dt} C_t = x_c C_t \left[-1 + y_c \frac{R_t}{R_t + R_0} \right] - x_p y_p \underbrace{\frac{P_t C_t}{C_t + C_0}}_{\substack{\text{functional response} \\ \text{(intake rate of predators)}}}$$

$$\frac{d}{dt} P_t = x_p P_t \left[-1 + y_p \frac{C_t}{C_t + C_0} \right]$$

$$\frac{dy}{dt} = f(y, t)$$

$$y_{t+1} = \int_t^{t+1} f(y_s, s) ds + y_t$$

Pros and cons

Pros and cons

Data-based models

Pros and cons

Data-based models

+ Demand little a priori knowledge

Pros and cons

Data-based models

- \+ Demand little a priori knowledge
- – Demand a large amount of data

Pros and cons

Data-based models

- + Demand little a priori knowledge
- – Demand a large amount of data
- – Limited interpretability (for ML models)

Pros and cons

Data-based models

+ Demand little a priori knowledge
– Demand a large amount of data
– Limited interpretability (for ML models)
– Limited extrapolability

Pros and cons

Data-based models

+ Demand little a priori knowledge
– Demand a large amount of data
– Limited interpretability (for ML models)
– Limited extrapolability

Pros and cons

Data-based models

- \+ Demand little a priori knowledge
- – Demand a large amount of data
- – Limited interpretability (for ML models)
- – Limited extrapolability

Pros and cons

Data-based models

- + Demand little a priori knowledge
- – Demand a large amount of data
- – Limited interpretability (for ML models)
- – Limited extrapolability

Process-based models

Pros and cons

Data-based models

+ Demand little a priori knowledge
– Demand a large amount of data
– Limited interpretability (for ML models)
– Limited extrapolability

Process-based models

+ Can **extrapolate**

Pros and cons

Data-based models

- \+ Demand little a priori knowledge
- – Demand a large amount of data
- – Limited interpretability (for ML models)
- – Limited extrapolability

Process-based models

- \+ Can **extrapolate**
- \+ **Interpretable**, can be extended, transferred, analytically understood

Pros and cons

Data-based models

- \+ Demand little a priori knowledge
- – Demand a large amount of data
- – Limited interpretability (for ML models)
- – Limited extrapolability

Process-based models

- \+ Can **extrapolate**
- \+ **Interpretable**, can be extended, transferred, analytically understood
- – Hard to calibrate

Pros and cons

Data-based models

- \+ Demand little a priori knowledge
- – Demand a large amount of data
- – Limited interpretability (for ML models)
- – Limited extrapolability

Process-based models

- \+ Can **extrapolate**
- \+ **Interpretable**, can be extended, transferred, analytically understood
- – Hard to calibrate
- – Suffer from inaccuracies, which make them less predictive than their data-based counterparts

I. From the mechanistic world to the ML world

Constraining NN with process-based models

Journal of Computational Physics 378 (2019) 686–707

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

ELSEVIER

Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations

M. Raissi [a], P. Perdikaris [b,*], G.E. Karniadakis [a]

[a] Division of Applied Mathematics, Brown University, Providence, RI, 02912, USA
[b] Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA, 19104, USA

Constraining NN with process-based models

Journal of Computational Physics 378 (2019) 686–707

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

ELSEVIER

Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations

M. Raissi [a], P. Perdikaris [b,*], G.E. Karniadakis [a]

[a] Division of Applied Mathematics, Brown University, Providence, RI, 02912, USA
[b] Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA, 19104, USA

PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Systems biology informed deep learning for inferring parameters and hidden dynamics

Alireza Yazdani [1], Lu Lu [2], Maziar Raissi [3], George Em Karniadakis [1*]

1 Division of Applied Mathematics, Brown University, Providence, Rhode Island, USA, **2** Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, **3** Department of Applied Mathematics, University of Colorado, Boulder, Colorado, USA

Constraining NN with process-based models

Journal of Computational Physics 378 (2019) 686–707

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations

M. Raissi[a], P. Perdikaris[b,*], G.E. Karniadakis[a]

[a] Division of Applied Mathematics, Brown University, Providence, RI, 02912, USA
[b] Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA, 19104, USA

PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Systems biology informed deep learning for inferring parameters and hidden dynamics

Alireza Yazdani[1☯], Lu Lu[2☯], Maziar Raissi[3], George Em Karniadakis[1*]

1 Division of Applied Mathematics, Brown University, Providence, Rhode Island, USA, 2 Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 3 Department of Applied Mathematics, University of Colorado, Boulder, Colorado, USA

Use scientific knowledge embedded in the available process-based model to **constrain a neural network**

Constraining NN with process-based models

Journal of Computational Physics 378 (2019) 686–707

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

ELSEVIER

Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations

M. Raissi[a], P. Perdikaris[b,*], G.E. Karniadakis[a]

[a] Division of Applied Mathematics, Brown University, Providence, RI, 02912, USA
[b] Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA, 19104, USA

PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Systems biology informed deep learning for inferring parameters and hidden dynamics

Alireza Yazdani[1], Lu Lu[2], Maziar Raissi[3], George Em Karniadakis[1*]

1 Division of Applied Mathematics, Brown University, Providence, Rhode Island, USA, **2** Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, **3** Department of Applied Mathematics, University of Colorado, Boulder, Colorado, USA

Use scientific knowledge embedded in the available process-based model to **constrain a neural network**

NN complies both with data and knowledge

$$\mathrm{NN}_\theta(t_i) \approx y_i \qquad \text{and} \qquad \frac{d}{dt}\mathrm{NN}_\theta(t) \approx f(\mathrm{NN}_\theta(t), t)$$

Constraining NN with process-based models

Journal of Computational Physics 378 (2019) 686–707

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations

M. Raissi [a], P. Perdikaris [b,*], G.E. Karniadakis [a]

[a] Division of Applied Mathematics, Brown University, Providence, RI, 02912, USA
[b] Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA, 19104, USA

PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Systems biology informed deep learning for inferring parameters and hidden dynamics

Alireza Yazdani [1], Lu Lu [2], Maziar Raissi [3], George Em Karniadakis [1*]

1 Division of Applied Mathematics, Brown University, Providence, Rhode Island, USA, 2 Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 3 Department of Applied Mathematics, University of Colorado, Boulder, Colorado, USA

Use scientific knowledge embedded in the available process-based model to **constrain a neural network**

NN complies both with data and knowledge

$$\mathrm{NN}_\theta(t_i) \approx y_i \qquad \text{and} \qquad \frac{d}{dt}\mathrm{NN}_\theta(t) \approx f(\mathrm{NN}_\theta(t), t)$$

Constraining NN with process-based models

Journal of Computational Physics 378 (2019) 686–707

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations

M. Raissi [a], P. Perdikaris [b,*], G.E. Karniadakis [a]

[a] Division of Applied Mathematics, Brown University, Providence, RI, 02912, USA
[b] Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA, 19104, USA

PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Systems biology informed deep learning for inferring parameters and hidden dynamics

Alireza Yazdani [1], Lu Lu [2], Maziar Raissi [3], George Em Karniadakis [1*]

1 Division of Applied Mathematics, Brown University, Providence, Rhode Island, USA, 2 Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 3 Department of Applied Mathematics, University of Colorado, Boulder, Colorado, USA

Use scientific knowledge embedded in the available process-based model to **constrain a neural network**

NN complies both with data and knowledge

$$\mathrm{NN}_\theta(t_i) \approx y_i \qquad \text{and} \qquad \frac{d}{dt}\mathrm{NN}_\theta(t) \approx f(\mathrm{NN}_\theta(t), t)$$

$$L(\theta) = L^{\mathrm{data}}(\theta) + L^{\mathrm{ODE}}(\theta, p)$$

Constraining NN with process-based models

Journal of Computational Physics 378 (2019) 686–707

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations

M. Raissi [a], P. Perdikaris [b,*], G.E. Karniadakis [a]

[a] Division of Applied Mathematics, Brown University, Providence, RI, 02912, USA
[b] Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA, 19104, USA

PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Systems biology informed deep learning for inferring parameters and hidden dynamics

Alireza Yazdani [1☯], Lu Lu [2☯], Maziar Raissi [3], George Em Karniadakis [1*]

1 Division of Applied Mathematics, Brown University, Providence, Rhode Island, USA, 2 Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 3 Department of Applied Mathematics, University of Colorado, Boulder, Colorado, USA

Use scientific knowledge embedded in the available process-based model to **constrain a neural network**

NN complies both with data and knowledge

$$\mathrm{NN}_\theta(t_i) \approx y_i \quad \text{and} \quad \frac{d}{dt}\mathrm{NN}_\theta(t) \approx f(\mathrm{NN}_\theta(t), t)$$

$$L(\theta) = L^{\mathrm{data}}(\theta) + L^{\mathrm{ODE}}(\theta, p)$$

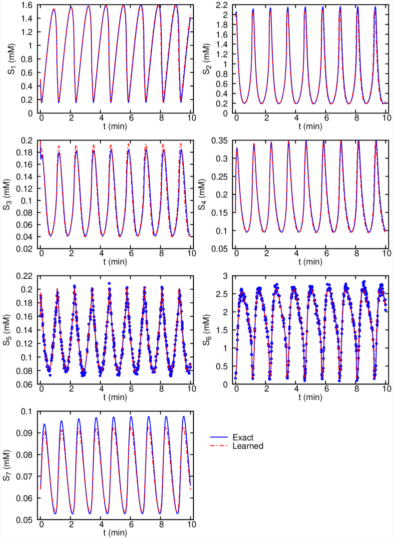
where

$$L^{\mathrm{ODE}}(\theta) = \sum_i \left|\left| \frac{d\,\mathrm{NN}_\theta(t_i)}{dt} - f(\mathrm{NN}_\theta(t_i), t_i) \right|\right|^2$$

Constraining NN with process-based models

Constraining NN with process-based models

NN_θ can predict variables for which it has never seen data!

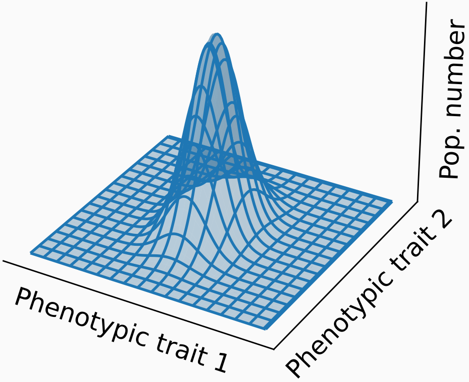


Yazdani et al., 2020

Using neural networks to solve high-dimensional PDEs

Using neural networks to solve high-dimensional PDEs

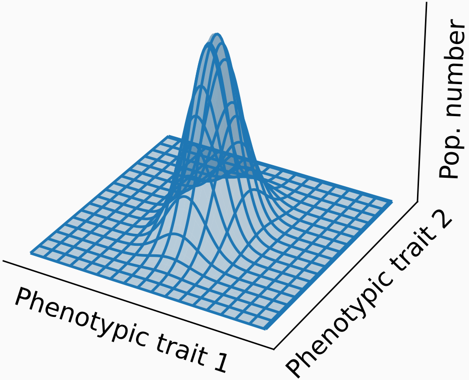
Modelling population number y as a function of continuous traits z



Using neural networks to solve high-dimensional PDEs

Modelling population number y as a function of continuous traits z

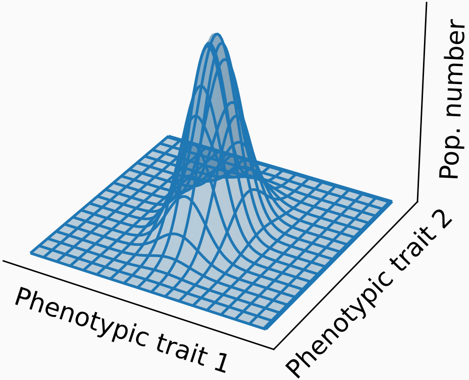
- population height



Using neural networks to solve high-dimensional PDEs

Modelling population number y as a function of continuous traits z

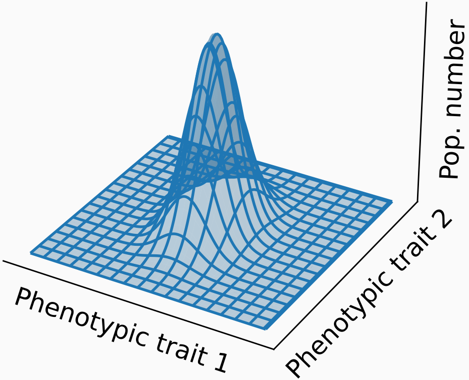
- population height
- population thermal niche



Using neural networks to solve high-dimensional PDEs

Modelling population number y as a function of continuous traits z

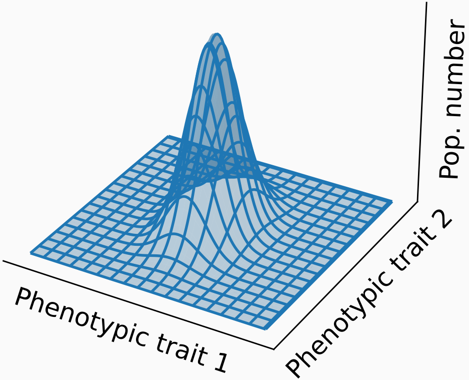
- population height
- population thermal niche
- population age
- …



Using neural networks to solve high-dimensional PDEs

Modelling population number y as a function of continuous traits z

- population height
- population thermal niche
- population age
- ...

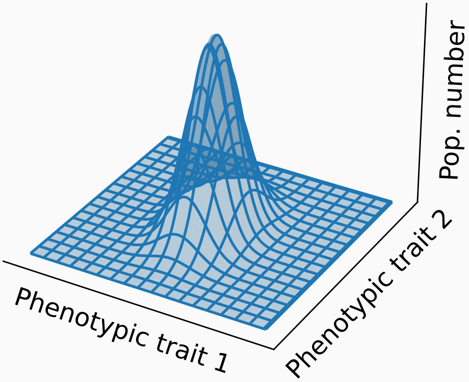


Using neural networks to solve high-dimensional PDEs

Modelling population number y as a function of continuous traits z

- population height
- population thermal niche
- population age
- ...

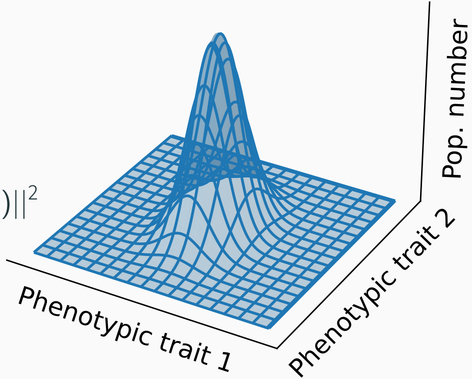
$$\frac{d}{dt} \underbrace{y(t,z)}_{\text{Population number for trait } z} = f(t, y, \partial_z y, \partial_{zz} y, \int y(t, \mathbf{z}) d\mathbf{z})$$



Using neural networks to solve high-dimensional PDEs

$$\mathrm{NN}(t, z) \approx x(t, z)$$

$$L^{\mathrm{ODE}}(\theta) = \sum_i \sum_j || \frac{d\,\mathrm{NN}(t_i, z_j)}{dt} - f(t_i, \mathrm{NN}(t_i, z_j), \ldots) ||^2$$



Curse of dimensionality

Curse of dimensionality

- **Computational complexity** of standard numerical schemes

Curse of dimensionality

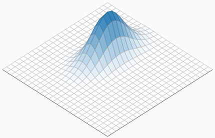
- **Computational complexity** of standard numerical schemes

$$z \in \mathbb{R}$$
$$\mathcal{O}(N)$$

Curse of dimensionality

- **Computational complexity** of standard numerical schemes

$z \in \mathbb{R}$	$z \in \mathbb{R}^2$
$\mathcal{O}(N)$	$\mathcal{O}(N^2)$



Curse of dimensionality

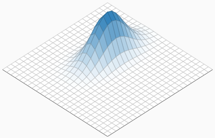
- **Computational complexity** of standard numerical schemes

$z \in \mathbb{R}$

$\mathcal{O}(N)$

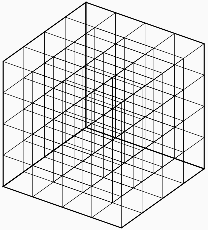
$z \in \mathbb{R}^2$

$\mathcal{O}(N^2)$



$z \in \mathbb{R}^3$

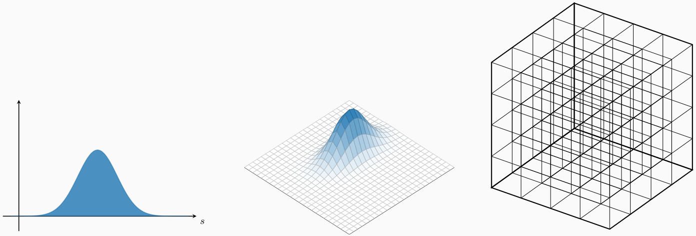
$\mathcal{O}(N^3)$



Curse of dimensionality

- **Computational complexity** of standard numerical schemes

$z \in \mathbb{R}$	$z \in \mathbb{R}^2$	$z \in \mathbb{R}^3$
$\mathcal{O}(N)$	$\mathcal{O}(N^2)$	$\mathcal{O}(N^3)$



- Standard numerical schemes for solving PDEs suffer the **curse of dimensionality**.

Mesh-free deep-learning methods for simulating high-dimensional models

Mesh-free deep-learning methods for simulating high-dimensional models

Machine learning-based method

Partial Differential Equations and Applications (2023) 4:51
https://doi.org/10.1007/s42985-023-00244-0

ORIGINAL PAPER

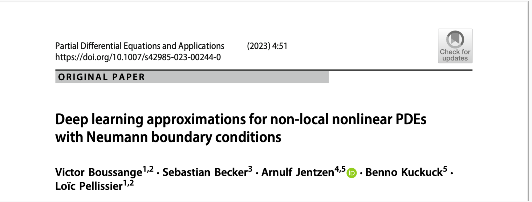
Deep learning approximations for non-local nonlinear PDEs with Neumann boundary conditions

Victor Boussange[1,2] · Sebastian Becker[3] · Arnulf Jentzen[4,5] · Benno Kuckuck[5] · Loïc Pellissier[1,2]

Mesh-free deep-learning methods for simulating high-dimensional models

Machine learning-based method

- Approximation of the solution with NNs
- NNs trained through **Monte Carlo approximation** of a **stochastic reformulation** of the PDE problem (Feynman–Kac)



Partial Differential Equations and Applications (2023) 4:51
https://doi.org/10.1007/s42985-023-00244-0

ORIGINAL PAPER

Deep learning approximations for non-local nonlinear PDEs with Neumann boundary conditions

Victor Boussange[1,2] · Sebastian Becker[3] · Arnulf Jentzen[4,5] · Benno Kuckuck[5] · Loïc Pellissier[1,2]

Intuition of mesh-free numerical methods

PDE Problem

$$\partial_t u(t,x) = \mu(t,x)\nabla_x u(t,x) + \frac{1}{2}\sigma^2(t,x)\Delta_x u(t,x)$$

with initial conditions $u(0,x) = g(x)$, where
$u\colon \mathbb{R}^d \to \mathbb{R}$.

Intuition of mesh-free numerical methods

PDE Problem

$$\partial_t u(t,x) = \mu(t,x)\nabla_x u(t,x) + \frac{1}{2}\sigma^2(t,x)\Delta_x u(t,x)$$

with initial conditions $u(0,x) = g(x)$, where $u\colon \mathbb{R}^d \to \mathbb{R}$.

Stochastic reformulation through Feynman–Kac formula

$$u(t,x) = \mathbb{E}\left[g(X_t^x)\right]$$

with X_t^x a stochastic process

$$X_t^x = \int_0^t \mu(X_s^x)ds + \int_0^t \sigma(X_s^x)dB_s + x.$$

Intuition of mesh-free numerical methods

PDE Problem

$$\partial_t u(t,x) = \mu(t,x)\nabla_x u(t,x) + \frac{1}{2}\sigma^2(t,x)\Delta_x u(t,x)$$

with initial conditions $u(0,x) = g(x)$, where $u\colon \mathbb{R}^d \to \mathbb{R}$.

Stochastic reformulation through Feynman–Kac formula

$$u(t,x) = \mathbb{E}\left[g(X_t^x)\right]$$

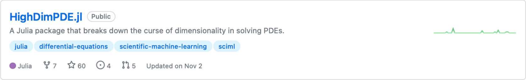
with X_t^x a stochastic process

$$X_t^x = \int_0^t \mu(X_s^x)ds + \int_0^t \sigma(X_s^x)dB_s + x.$$

Monte Carlo approximation

$$u(t,x) \approx \frac{1}{N}\sum_i g(X_t^x)$$

`HighDimPDE.jl`: A package implementing recent solver algorithms that break down the curse of dimensionality



`HighDimPDE.jl`: A package implementing recent solver algorithms that break down the curse of dimensionality

HighDimPDE.jl Public
A Julia package that breaks down the curse of dimensionality in solving PDEs.
julia differential-equations scientific-machine-learning sciml
Julia 7 60 4 5 Updated on Nov 2

`HighDimPDE.jl` belongs to the SciML ecosystem

SciML Open Source Scientific Machine Learning
Open source software for scientific machine learning
904 followers https://sciml.ai @SciML_Org contact@chrisrackauckas.com

`HighDimPDE.jl`: A package implementing recent solver algorithms that break down the curse of dimensionality

HighDimPDE.jl Public

A Julia package that breaks down the curse of dimensionality in solving PDEs.

julia differential-equations scientific-machine-learning sciml

Julia 7 60 4 5 Updated on Nov 2

`HighDimPDE.jl` belongs to the SciML ecosystem

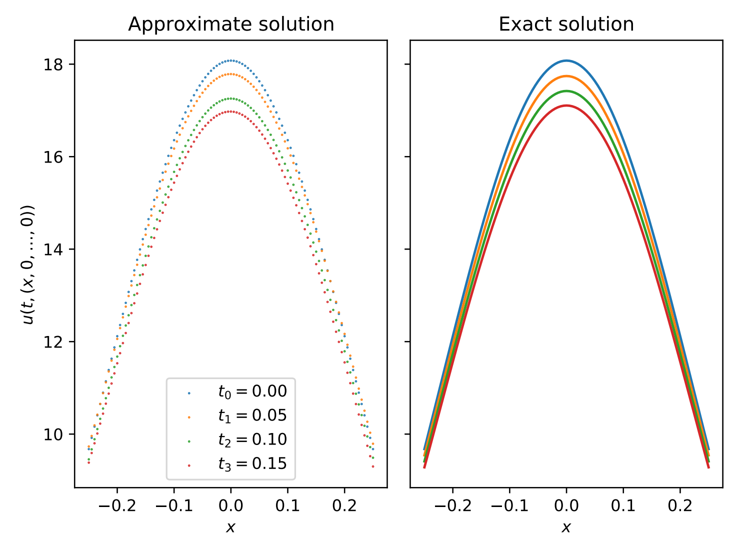
SciML Open Source Scientific Machine Learning

Open source software for scientific machine learning

904 followers https://sciml.ai @SciML_Org contact@chrisrackauckas.com

```julia
using HighDimPDE
alg = DeepSplitting(kwargs...)
prob = PIDEProblem(kwargs...)
sol = solve(prob, alg, kwargs...)
```

We are now able to simulate **10-dimensional eco-evolutionary models**!



- Scientific Machine Learning

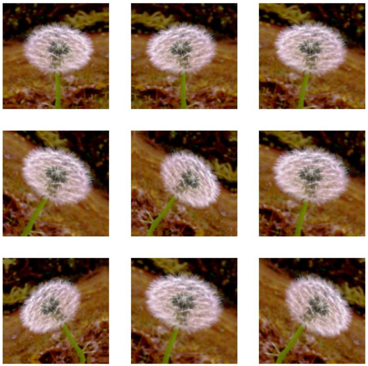
- Scientific Machine Learning
- We can constrain NNs with ecological knowledge by adding additional constraints in the loss function

- Scientific Machine Learning
- We can constrain NNs with ecological knowledge by adding additional constraints in the loss function
- Not only can physics-informed NNs facilitate data assimilation, but they can facilitate the simulation of high dimensional process-based models

Using ecological knowledge to augment data for the training of a NN

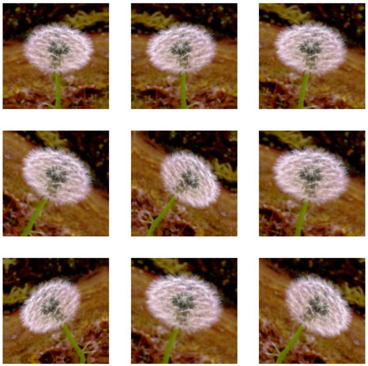
Using ecological knowledge to augment data for the training of a NN

```
data_augmentation = keras.Sequential(
[
    layers.RandomFlip("horizontal",
    input_shape=(img_height,
    img_width,
    3)),
    layers.RandomRotation(0.1),
    layers.RandomZoom(0.1),
  ]
)
```



Using ecological knowledge to augment data for the training of a NN

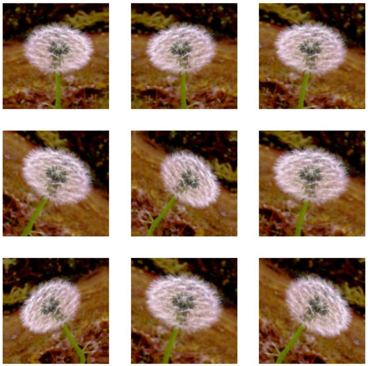
```
data_augmentation = keras.Sequential(
[
    layers.RandomFlip("horizontal",
    input_shape=(img_height,
    img_width,
    3)),
    layers.RandomRotation(0.1),
    layers.RandomZoom(0.1),
  ]
)
```



- An image of a flower will still be an image of a flower under small rotation, flip, and zooming

Using ecological knowledge to augment data for the training of a NN

```
data_augmentation = keras.Sequential(
[
    layers.RandomFlip("horizontal",
    input_shape=(img_height,
    img_width,
    3)),
    layers.RandomRotation(0.1),
    layers.RandomZoom(0.1),
  ]
)
```



- An image of a flower will still be an image of a flower under small rotation, flip, and zooming
- Augmenting data helps the ML model to **generalize better**

Attribution of biodiversity change to climate change and land-use

Attribution of biodiversity change to climate change and land-use

natural climate variations 🌦️

anthropogenic climate change ✈️

changes in land use and land cover 🌐

🦋 biodiversity change

Attribution of biodiversity change to climate change and land-use

?

natural climate variations 🌦️

anthropogenic climate change ✈️ 🦋 biodiversity change

changes in land use and land cover 🌐

Building a macro-ecological model accounting for habitat area

Species-Area models have been central
to predict extinctions due to habitat loss

Building a macro-ecological model accounting for habitat area

Species-Area models have been central to predict extinctions due to habitat loss

Extinction risk from climate change

Chris D. Thomas[1], Alison Cameron[1], Rhys E. Green[2], Michel Bakkenes[3], Linda J. Beaumont[4], Yvonne C. Collingham[5], Barend F. N. Erasmus[6], Marinez Ferreira de Siqueira[7], Alan Grainger[8], Lee Hannah[9], Lesley Hughes[4], Brian Huntley[5], Albert S. van Jaarsveld[10], Guy F. Midgley[11], Lera Miles[8*], Miguel A. Ortega-Huerta[12], A. Townsend Peterson[13], Oliver L. Phillips[8] & Stephen E. Williams[14]

Building a macro-ecological model accounting for habitat area

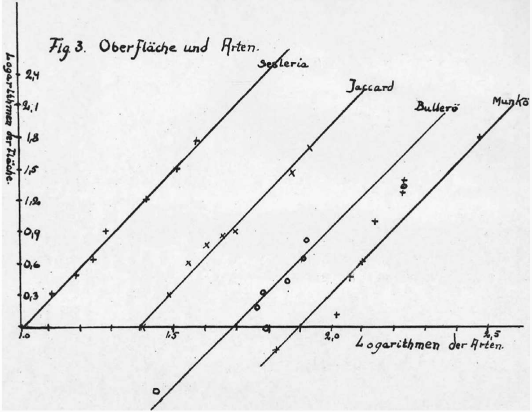
Species-Area models have been central to predict extinctions due to habitat loss

$$\log SR = \overbrace{\log c}^{\text{intercept}} + \underbrace{z}_{\text{slope}} \log A$$

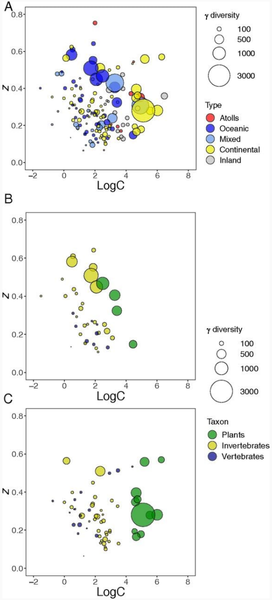
Extinction risk from climate change

Chris D. Thomas[1], Alison Cameron[1], Rhys E. Green[2], Michel Bakkenes[3], Linda J. Beaumont[4], Yvonne C. Collingham[5], Barend F. N. Erasmus[6], Marinez Ferreira de Siqueira[7], Alan Grainger[8], Lee Hannah[9], Lesley Hughes[4], Brian Huntley[5], Albert S. van Jaarsveld[10], Guy F. Midgley[11], Lera Miles[8*], Miguel A. Ortega-Huerta[12], A. Townsend Peterson[13], Oliver L. Phillips[8] & Stephen E. Williams[14]

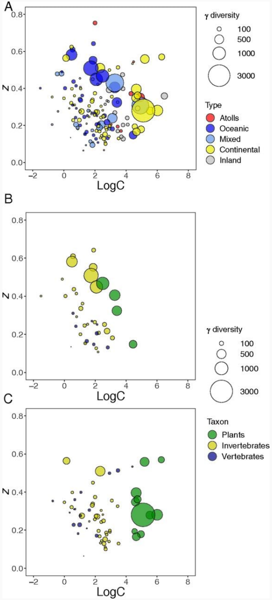
$$\overbrace{SR}^{\text{Species richness}} = c \underbrace{A^z}_{\text{Habitat area}}$$



Building a macro-ecological model accounting for habitat area



Building a macro-ecological model accounting for habitat area

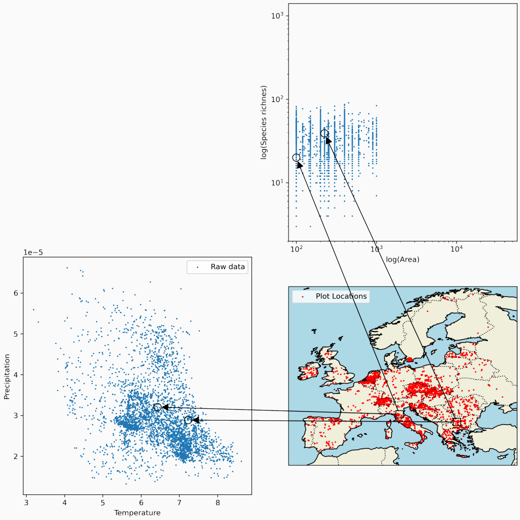


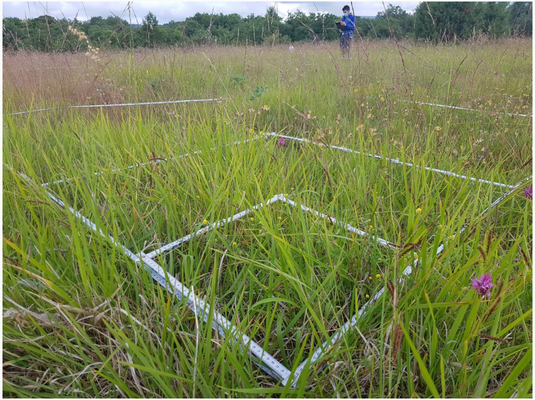
💡

$$SR = c(u)A^{z(u)}$$

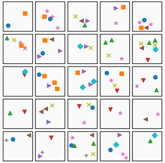
where u are additional features corresponding to **environmental conditions**

Building a macro-ecological model accounting for habitat area

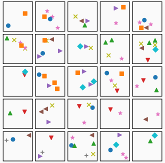




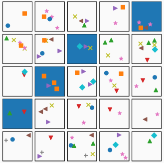
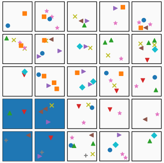
Building a macro-ecological model accounting for habitat area



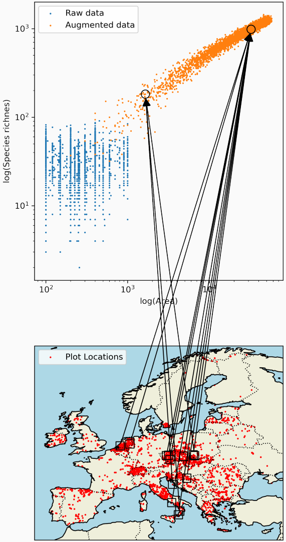
Building a macro-ecological model accounting for habitat area



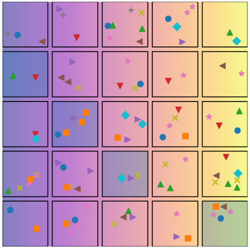
As a first approximation

$\mathbb{E}\Big[\mathrm{SR}\Big($  $\Big)\Big] = \mathbb{E}\Big[\mathrm{SR}\Big($  $\Big)\Big]$

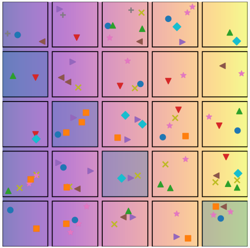
Building a macro-ecological model accounting for habitat area



Building a macro-ecological model accounting for habitat area



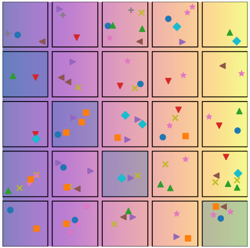
Building a macro-ecological model accounting for habitat area



How to account for

- Spatial auto-correlation (limited dispersal)

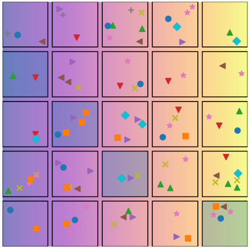
Building a macro-ecological model accounting for habitat area



How to account for

- Spatial auto-correlation (limited dispersal)
- Environmental heterogeneity

Building a macro-ecological model accounting for habitat area

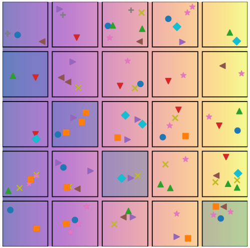


How to account for

- Spatial auto-correlation (limited dispersal)
- Environmental heterogeneity

?

Building a macro-ecological model accounting for habitat area



How to account for

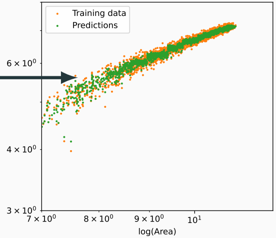
- Spatial auto-correlation (limited dispersal)
- Environmental heterogeneity

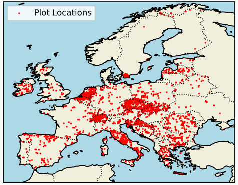
?

$$\hat{y} = \mathcal{M}\left(A, \begin{array}{c} \begin{array}{ccc}\text{Site 1} & \quad\quad & \text{Site 4}\end{array} \\ \begin{pmatrix} u_{11} & & u_{14} \\ & \vdots & \\ u_{21} & & u_{24} \\ u_{31} & & u_{34} \end{pmatrix}\end{array}, \begin{array}{c} \begin{array}{ccc}\text{Site 1} & \quad\quad & \text{Site 4}\end{array} \\ \begin{pmatrix} d_{11} & & d_{14} \\ & \vdots & \\ d_{21} & & d_{24} \\ d_{31} & & d_{34} \end{pmatrix}\end{array}\right)$$

Building a macro-ecological model accounting for habitat area

$$\hat{y} = \mathcal{M}\left(A, \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \end{pmatrix}, \begin{pmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{pmatrix}\right)$$

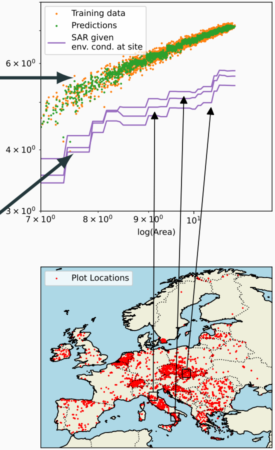




Building a macro-ecological model accounting for habitat area

$$\hat{y} = \mathcal{M}\left(A, \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \end{pmatrix}, \begin{pmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{pmatrix}\right)$$

$$\hat{y} = \mathcal{M}\left(A, \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \end{pmatrix}, \begin{pmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{pmatrix}\right)$$

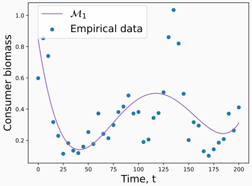


- We can augment the dataset used to train a ML model with ecological knowledge

From the ML world to the mechanistic world

After all, process-based models can be seen as regressors $\mathcal{M}_\theta$!

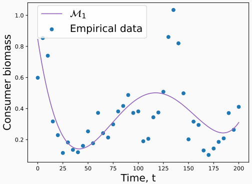
$$\frac{d}{dt}y = f_p(y, t)$$



After all, process-based models can be seen as regressors $\mathcal{M}_\theta$!

$$\frac{d}{dt}y = f_p(y, t)$$

$$\hat{y} = \underbrace{\int_{t_0}^{t} f_p(y_s, s)ds}_{\text{Numerical integration of the model}} + y_0$$



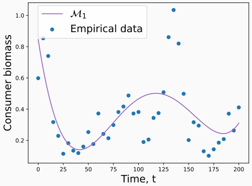
After all, process-based models can be seen as regressors $\mathcal{M}_\theta$!

$$\frac{d}{dt}y = f_p(y, t)$$

$$\hat{y} = \underbrace{\int_{t_0}^{t} f_p(y_s, s)ds}_{\text{Numerical integration of the model}} + y_0$$

$$= \mathcal{M}_\theta(t)$$

where $\theta = (y_0, p)$



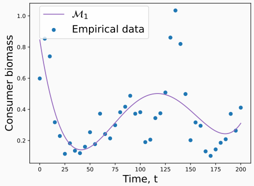
After all, process-based models can be seen as regressors $\mathcal{M}_\theta$!

$$\frac{d}{dt}y = f_p(y,t)$$

$$\hat{y} = \underbrace{\int_{t_0}^{t} f_p(y_s, s)ds}_{\text{Numerical integration of the model}} + y_0$$

$$= \mathcal{M}_\theta(t)$$

$$\text{where } \theta = (y_0, p)$$



- In principle, process-based models can be trained similarly to ML models

$$L(\theta, \mathbf{y}) = \sum_{k=1}^{K} ||y_k - \mathcal{M}_\theta(x_k)||^2$$

Available methods to minimize L

- Global optimization
  - Markov Chain Monte Carlo
  - Approximate Bayesian Computation
- Local optimization
  - Variational optimizers

Global optimization with MCMC

- Sample L with a Markov chain $\theta^1, \theta^2, \ldots$ which equilibrium distribution is proportional to L

Global optimization with MCMC

- Sample L with a Markov chain $\theta^1, \theta^2, \ldots$ which equilibrium distribution is proportional to L
- Estimate $\hat{\theta}$ and associated uncertainty based on the samples at equilibrium

Global optimization with MCMC

- Sample L with a Markov chain $\theta^1, \theta^2, \ldots$ which equilibrium distribution is proportional to L
- Estimate $\hat{\theta}$ and associated uncertainty based on the samples at equilibrium

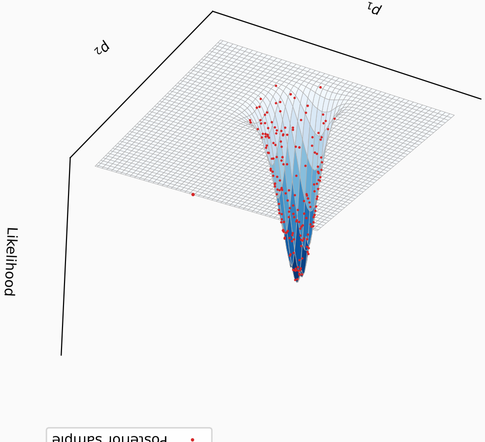
+ Provide uncertainty estimations

Global optimization with MCMC

- Sample L with a Markov chain $\theta^1, \theta^2, \ldots$ which equilibrium distribution is proportional to L
- Estimate $\hat{\theta}$ and associated uncertainty based on the samples at equilibrium

+ Provide uncertainty estimations
− Suffer from the curse of dimensionality

❌ Not suited for training complex process-based models

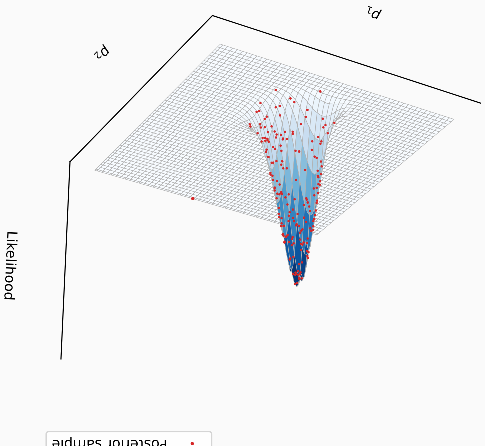


Global optimization with MCMC

- Sample L with a Markov chain $\theta^1, \theta^2, \ldots$ which equilibrium distribution is proportional to L
- Estimate $\hat{\theta}$ and associated uncertainty based on the samples at equilibrium

+ Provide uncertainty estimations
− Suffer from the curse of dimensionality
− Process-based models are costly to evaluate

❌ Not suited for training complex process-based models

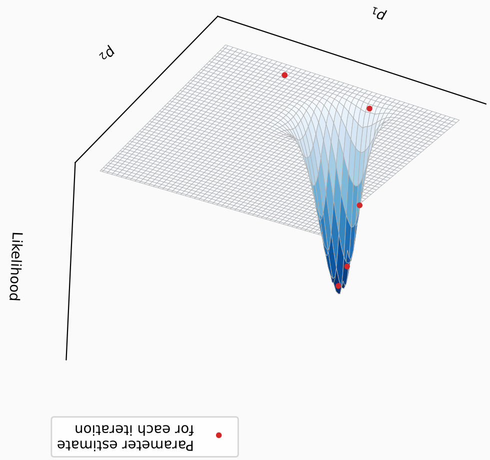


Local optimization with gradient descent

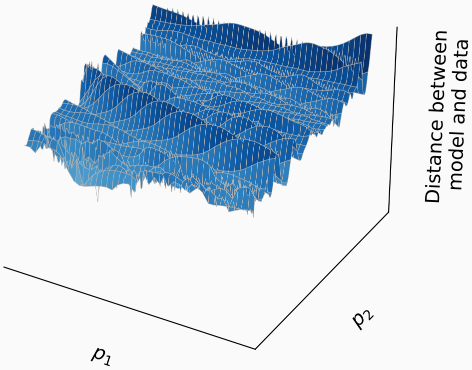
- Follow the steepest slope $\nabla_\theta L(\theta, \mathbf{y})$

$$\theta^{m+1} = \underbrace{\theta^m}_{\text{parameter at iteration } m} - \overbrace{\lambda}^{\text{learning rate}} \underbrace{\nabla_\theta L(\theta^{(m)}, \mathbf{y})}_{\text{gradient w.r.t parameters}}$$

+Less prone to the curse of dimensionality
−Parameter point estimates
−Convergence to local minima
−Require parameter sensitivity

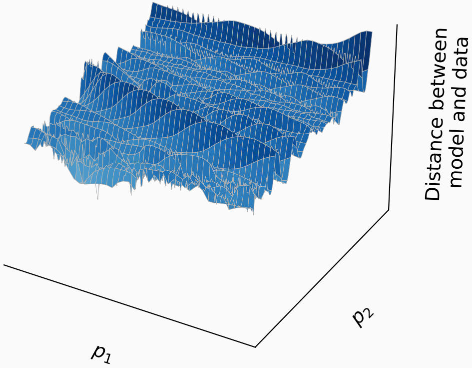


The specificities of ecological models



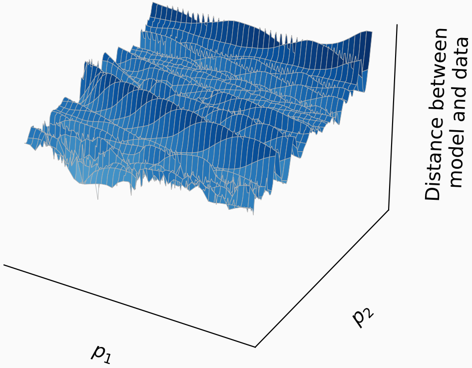
The specificities of ecological models

❌ Forward pass is expensive



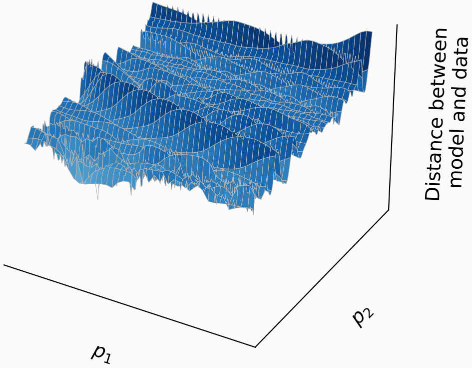
The specificities of ecological models

❌ Forward pass is expensive



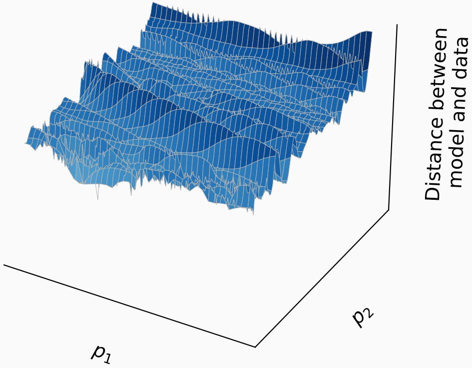
The specificities of ecological models

- ❌ Forward pass is expensive
- ❌ Many local minima



The specificities of ecological models

- ❌ Forward pass is expensive
- ❌ Many local minima
- ❌ Require the sensitivity to the model parameters, $\nabla_\theta \mathcal{M}_\theta$

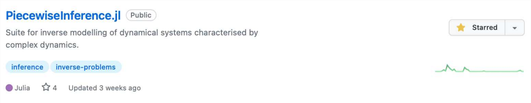


`PiecewiseInference.jl`: Inverse modelling framework for dynamical systems with highly non-linear dynamics.

Boussange, V., Vilimelis-Aceituno, P., Schäfer, F., Pellissier, L., *Partitioning ecological time series to improve process-based models with machine learning* [bioRxiv] (2022), 46 pages. In review.

`PiecewiseInference.jl`: Inverse modelling framework for dynamical systems with highly non-linear dynamics.

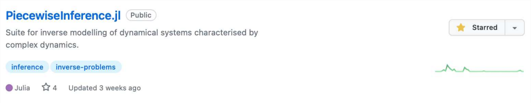
- **segmentation method** with minibatches



Boussange, V., Vilimelis-Aceituno, P., Schäfer, F., Pellissier, L., *Partitioning ecological time series to improve process-based models with machine learning* [bioRxiv] (2022), 46 pages. In review.

`PiecewiseInference.jl`: Inverse modelling framework for dynamical systems with highly non-linear dynamics.

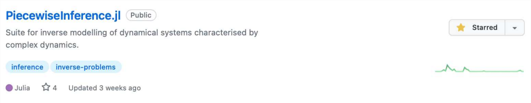
- **segmentation method** with minibatches
- sensitivity analysis methods based on **Automatic Differentiation**



Boussange, V., Vilimelis-Aceituno, P., Schäfer, F., Pellissier, L., *Partitioning ecological time series to improve process-based models with machine learning* [bioRxiv] (2022), 46 pages. In review.

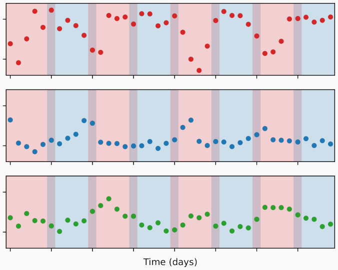
`PiecewiseInference.jl`: Inverse modelling framework for dynamical systems with highly non-linear dynamics.

- **segmentation method** with minibatches
- sensitivity analysis methods based on **Automatic Differentiation**
- use of **deep learning variational optimizers**

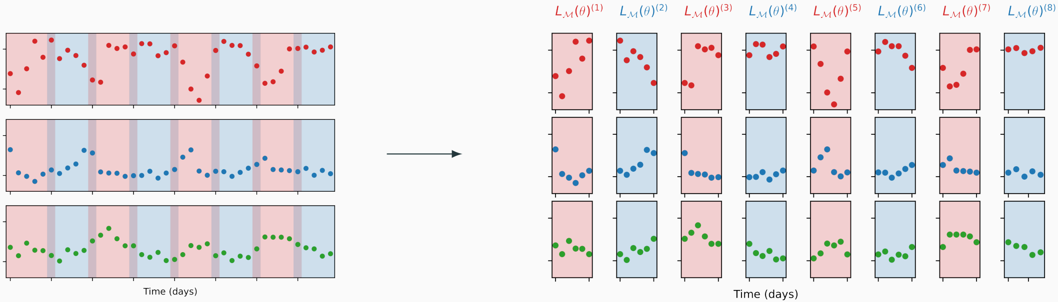


Boussange, V., Vilimelis-Aceituno, P., Schäfer, F., Pellissier, L., *Partitioning ecological time series to improve process-based models with machine learning* [bioRxiv] (2022), 46 pages. In review.

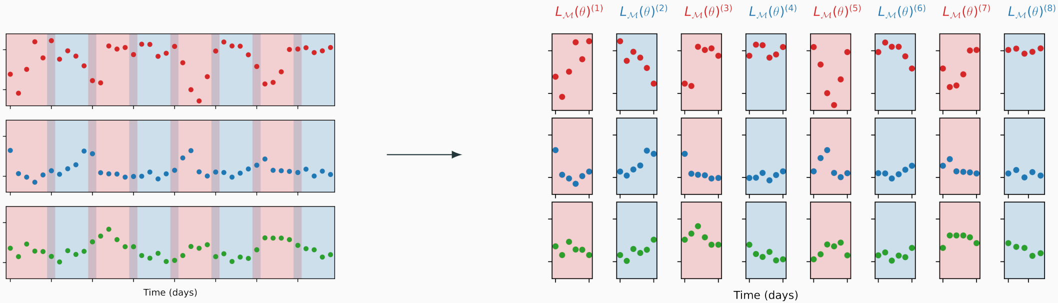
`PiecewiseInference.jl`: Inverse modelling framework for dynamical systems with highly non-linear dynamics.



`PiecewiseInference.jl`: Inverse modelling framework for dynamical systems with highly non-linear dynamics.

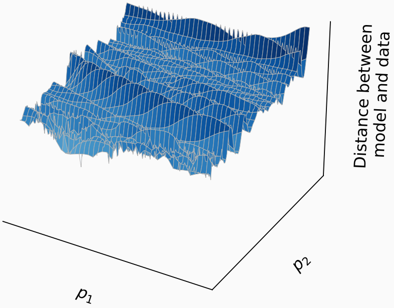


`PiecewiseInference.jl`: Inverse modelling framework for dynamical systems with highly non-linear dynamics.

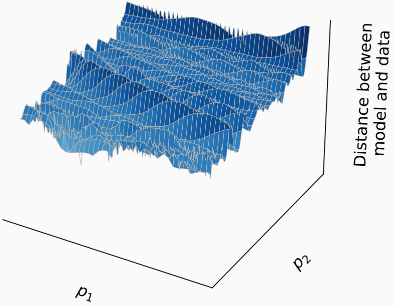


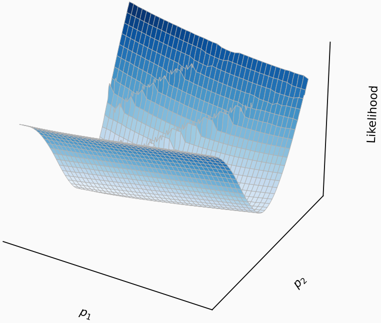
$$L_{\mathcal{M}}(\theta) = L_{\mathcal{M}}^{(1)}(\theta) + L_{\mathcal{M}}^{(2)}(\theta) + \dots \tag{1}$$

`PiecewiseInference.jl`: Inverse modelling framework for dynamical systems with highly non-linear dynamics.



`PiecewiseInference.jl`: Inverse modelling framework for dynamical systems with highly non-linear dynamics.





`PiecewiseInference.jl`: Inverse modelling framework for dynamical systems with highly non-linear dynamics.

- Deep learning optimizers

Adam: A method for stochastic optimization [PDF] arxiv.org
DP Kingma, J Ba - arXiv preprint arXiv:1412.6980, 2014 - arxiv.org
… **Adam** works well in practice and compares favorably to other stochastic optimization methods.
Finally, we discuss AdaMax, a variant of **Adam** … Overall, we show that **Adam** is a versatile …
Save Cite Cited by 161022 Related articles All 27 versions

`PiecewiseInference.jl`: Inverse modelling framework for dynamical systems with highly non-linear dynamics.

- Deep learning optimizers

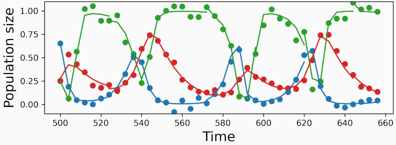
Adam: A method for stochastic optimization [PDF] arxiv.org
DP Kingma, J Ba - arXiv preprint arXiv:1412.6980, 2014 - arxiv.org
… **Adam** works well in practice and compares favorably to other stochastic optimization methods. Finally, we discuss AdaMax, a variant of **Adam** … Overall, we show that **Adam** is a versatile …
☆ Save 99 Cite Cited by 161022 Related articles All 27 versions

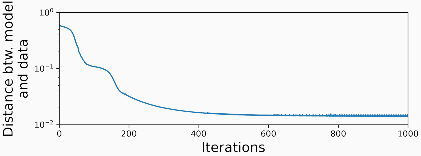
- **Sensitivity analysis** based on **automatic differentiation**

```julia
using ForwardDiff
ForwardDiff.gradient(sin, 0.1) == cos(0.1) # true
```

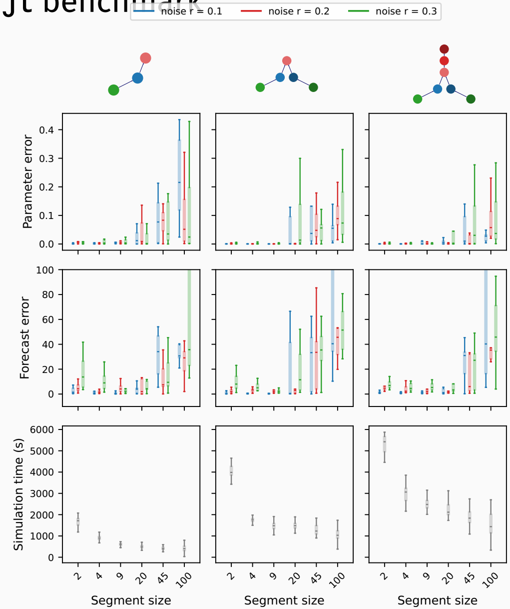
`PiecewiseInference.jl`: Inverse modelling framework for dynamical systems with highly non-linear dynamics.

```julia
using PiecewiseInference

model = MyModel(ModelParams(...))

infprob = InferenceProblem(model, p_init)

res = inference(infprob,
group_nb = 2,
data,
tsteps = tsteps,
epochs = [5000],
optimizers = [ADAM(0.001)],
batchsizes = [1])
```

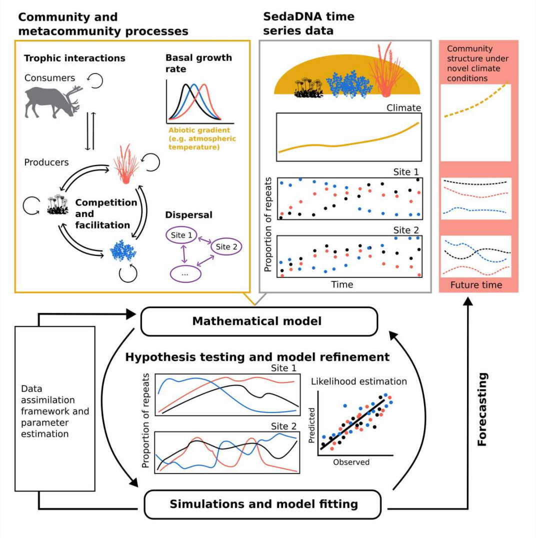




PiecewiseInference.jl benchmark

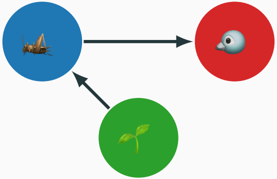


Dynamic forecast of future changes

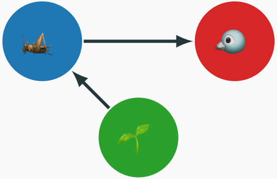


Alsos, I.G., **Boussange, V.**, ... *Using ancient sedimentary DNA to forecast ecosystem trajectories under climate change* (2023). Accepted in Philosophical Transactions of the Royal Society B

Neural network-based parametrization



Neural network-based parametrization

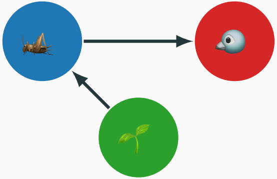


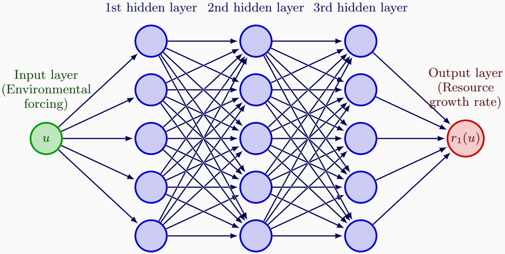
net growth rate 🌱 = basal growth(environmental conditions) − competition − grazing − death

net rowth rate 🦗 = grazing − predation − death

net growth rate 🐦 = predation − death

Neural network-based parametrization



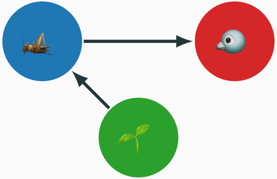


net growth rate 🌱 = NN(environmental conditions) − competition − grazing − death

net rowth rate 🦗 = grazing − predation − death

net growth rate 🐦 = predation − death

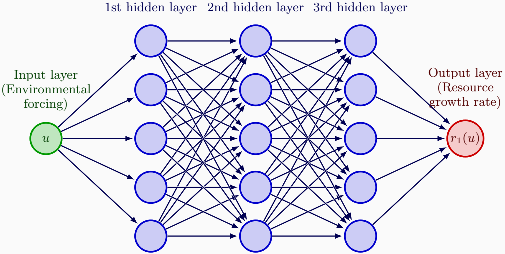
Neural network-based parametrization



$$\frac{d}{dt}R_t = \mathrm{NN(u)}\, \overbrace{R_t(1 - R_t)}^{\text{logistic growth}} - x_c y_c \underbrace{\frac{C_t R_t}{R_t + R_0}}_{\substack{\text{functional response} \\ \text{(intake rate of consumers)}}}$$

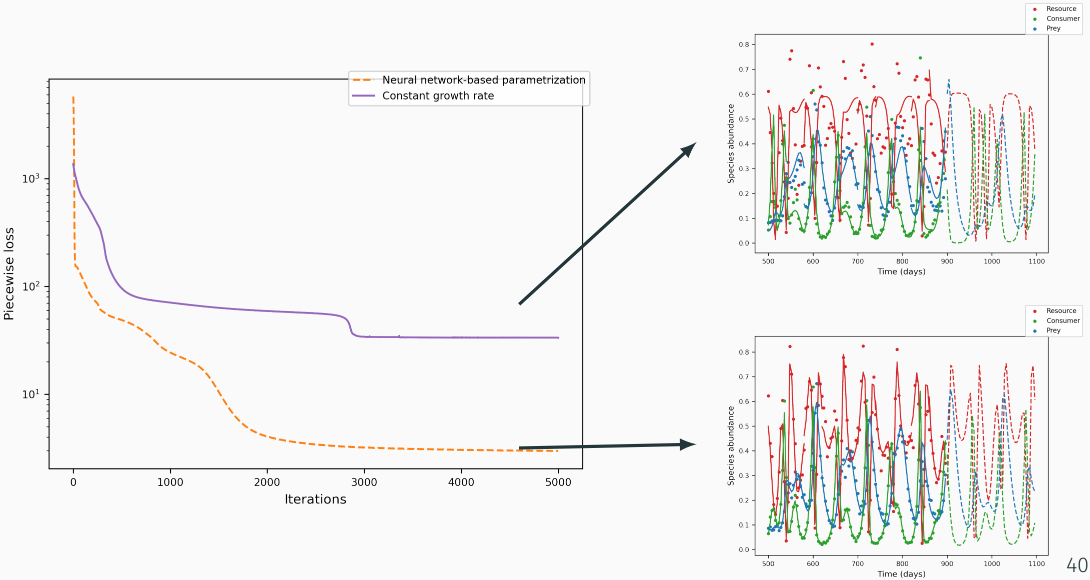
$$\frac{d}{dt}C_t = x_c C_t \left[-1 + y_c \frac{R_t}{R_t + R_0}\right] - x_p y_p \underbrace{\frac{P_t C_t}{C_t + C_0}}_{\substack{\text{functional response} \\ \text{(intake rate of predators)}}}$$

$$\frac{d}{dt}P_t = x_p P_t \left[-1 + y_p \frac{C_t}{C_t + C_0}\right]$$

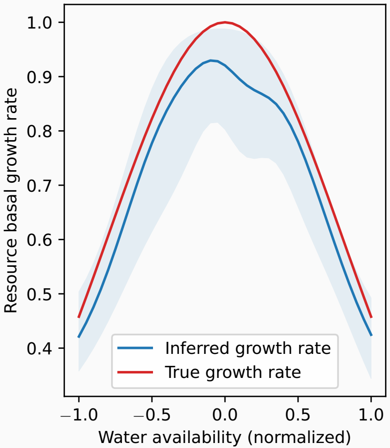


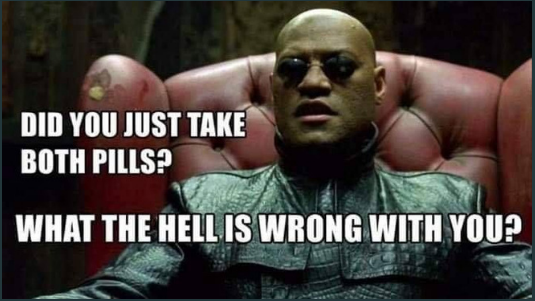
$$\theta = (x_c, y_c, x_p, y_p, C_0, R_0, \underbrace{\mathrm{W}_1, \mathrm{W}_2, \mathrm{W}_3, \mathrm{b}_1, \mathrm{b}_2, \mathrm{b}_3}_{\substack{\text{Weight and biases} \\ \text{of neural network}}})$$

Neural network-based parametrization



Offline interpretation of the neural network-based parametrization




DID YOU JUST TAKE
BOTH PILLS?
WHAT THE HELL IS WRONG WITH YOU?

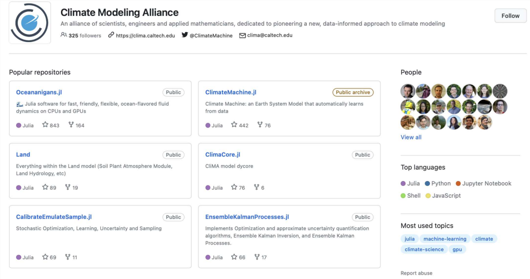
Paradigm shift

Paradigm shift

github.com/vboussange/WSLJuliaWorkshop2023

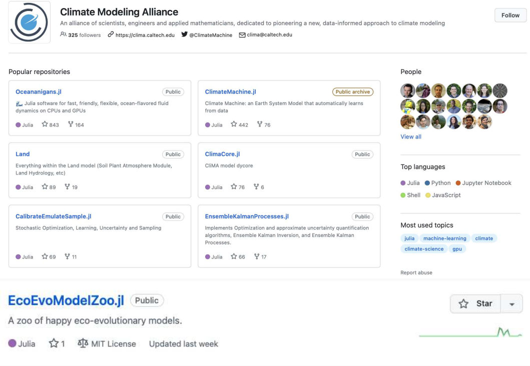
Paradigm shift

github.com/vboussange/WSLJuliaWorkshop2023



Paradigm shift

github.com/vboussange/WSLJuliaWorkshop2023



Summary

- Ecological knowledge can be used to inform ML models by

Summary

- Ecological knowledge can be used to inform ML models by
    - **Augmenting data** with ecological knowledge

Summary

- Ecological knowledge can be used to inform ML models by
  - **Augmenting data** with ecological knowledge
  - **Constraining ML models with process-based constraints**

Summary

- Ecological knowledge can be used to inform ML models by
  - **Augmenting data** with ecological knowledge
  - **Constraining ML models with process-based constraints**
- Physics-informed neural networks can **greatly accelerate process-based simulations**
- ML techniques can be transferred to benefit process-based approaches

Summary

- Ecological knowledge can be used to inform ML models by
  - **Augmenting data** with ecological knowledge
  - **Constraining ML models with process-based constraints**
- Physics-informed neural networks can **greatly accelerate process-based simulations**
- ML techniques can be transferred to benefit process-based approaches
  - **PiecewiseInference.jl**, a tool for inverse modeling in ecological systems with nonlinear dynamics

Summary

- Ecological knowledge can be used to inform ML models by
  - **Augmenting data** with ecological knowledge
  - **Constraining ML models with process-based constraints**
- Physics-informed neural networks can **greatly accelerate process-based simulations**
- ML techniques can be transferred to benefit process-based approaches
  - **PiecewiseInference.jl**, a tool for inverse modeling in ecological systems with nonlinear dynamics
  - **Neural-network based parametrization**

Summary

- Ecological knowledge can be used to inform ML models by
  - **Augmenting data** with ecological knowledge
  - **Constraining ML models with process-based constraints**
- Physics-informed neural networks can **greatly accelerate process-based simulations**
- ML techniques can be transferred to benefit process-based approaches
  - **PiecewiseInference.jl**, a tool for inverse modeling in ecological systems with nonlinear dynamics
  - **Neural-network based parametrization**
- We need a **programming paradigm shift** to levergage Scientific Machine Learning